144141

JPRS 83044

10 March 1983

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

No. 68
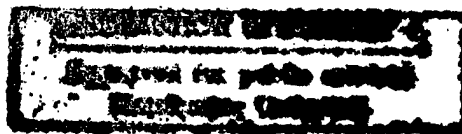
FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service (NTIS), Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semimonthly by the NTIS, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

JPRS REPORTS

Japan Report
Korean Affairs Report
Southeast Asia Report
Mongolia Report

Near East/South Asia Report
Sub-Saharan Africa Report
West Europe Report
West Europe Report: Science and Technology

Latin America Report

USSR

Political and Sociological Affairs
Problems of the Far East
Science and Technology Policy
Sociological Studies
Translations from KOMMUNIST
USA: Economics, Politics, Ideology
World Economy and International Relations
Agriculture
Construction and Related Industries
Consumer Goods and Domestic Trade
Economic Affairs
Energy
Human Resources
International Economic Relations
Transportation

Physics and Mathmetics
Space
Space Biology and Aerospace Medicine
Military Affairs
Chemistry
Cybernetics, Computers and Automation Technology
Earth Sciences
Electronics and Electrical Engineering
Engineering and Equipment
Machine Tools and Metal-Working Equipment
Life Sciences: Biomedical and Behavioral Sciences
Life Sciences: Effects of Nonionizing Electromagnetic
Radiation
Materials Science and Metallurgy

EASTERN EUROPE

Political, Sociological and Military Affairs
Scientific Affairs

Economic and Industrial Affairs

CHINA

Political, Sociological and Military Affairs
Economic Affairs
Science and Technology

RED FLAG
Agriculture
Plant and Installation Data

WORLDWIDE

Telecommunications Policy, Research and
  Development
Nuclear Development and Proliferation

Environmental Quality
Epidemiology

FBIS DAILY REPORT

China
Soviet Union
South Asia
Asia and Pacific

Eastern Europe
Western Europe
Latin America
Middle East and Africa

To order, see inside front cover

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

No. 68

CONTENTS

SOFTWARE

CONFERENCES

ORGANIZATIONS

PUBLICATIONS

COOPERATION OF CEMA NATIONS IN COMPUTER FIELD

Moscow EKONOMICHESKOYE SOTRUDNICHESTVO STRAN-CHLENOV SEV in Russian No 2, Feb 81 pp 20-25

[Article by Yakov Ryabov, first deputy chairman of Gosplan SSSR, permanent chairman of Intergovernmental Commission on Cooperation of Socialist Nations in the Field of Computer Technology: "Cooperation of Socialist Nations in the Field of Computer Technology: Status and Prospects"]

[Text] Eleven years have passed since the nations of socialist friendship-- Bulgaria, Hungary, East Germany, Poland, the Soviet Union and Czechoslovakia-- signed an agreement on cooperation in the field of development, production and use of facilities of computer technology. On the basis of this agreement, the Intergovernmental Commission was set up and charged with supervising and coordinating cooperation. Later the agreement was joined by the Republic of Cuba and Rumania.

The results of the intermittent period have demonstrated the decisive advantages of socialist economic integration, enabling mutually profitable cooperation in such a promising sector as computer technology.

Socialist nations have taken the path of broad cooperation in setting up new technological processes, and in producing highly efficient large-capacity machines and equipment. Statements of the brotherly parties on the need for further intense social production and improvement of production efficiency have necessitated extensive measures on perfecting management. The course taken herewith toward international integration and specialization within the framework of socialist cooperation has once again demonstrated to the world the enormous advantages of international socialist division of labor. As pointed out by Comrade L. I. Brezhnev at the Twenty-Sixth CPSU Congress, now it is impossible to imagine the sure development of any socialist nation, the successful resolution of problems such as introducing the latest advances of science and engineering without ties with other brotherly nations.

Today computer technology is one of the powerful factors ensuring improvement of efficiency of socialist production, a basis for development of automated control systems in leading areas of the economy. With each year its role increases in scientific research, planning, design and technological work. Computer technology is finding wide application in industry, agriculture,

1

Yakov Ryabov
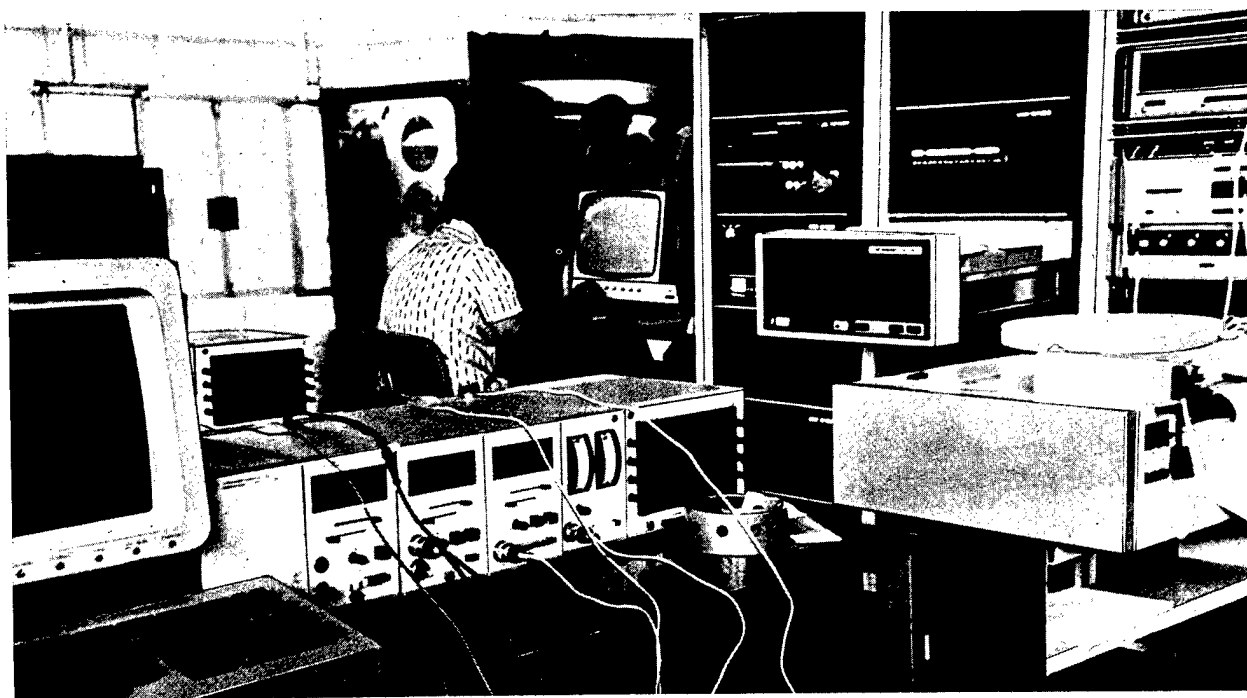
Information-measurement complex based on SM EVM system

construction, commerce, public health, and in the area of everyday services. It gives us a new way of solving social problems as well.

As practice shows, further development of the production potential of nations, an increase in the volumes of their social production will be determined increasingly by the capacity of the computer pool, its structure and parameters, the quality and reliability of peripheral equipment, and the level of development of software.

At the present time, computer technology, like no other technology, is a catalyst of scientific advances. And computer technology in turn is accumulating practically all advances in technical progress both in organization of production and in the development of totally new technological processes, materials, component parts and other forms of equipment. Qualitative shifts in a number of subsectors of metallurgy, chemistry, electrical engineering, the electronics industry and instrument making are due in large measure to the increasing requirements for materials and component parts on the part of computer facilities that are being produced. The development of computer software is becoming a large industry.

All this has made it necessary to consolidate the forces and funds of socialist nations in this field, has demanded the development of a unified technical and matched economic policy, in other words, international specialization and integration of effective large-series production of computers and peripheral equipment.

Accepted concepts that have been discussed with the participation of skilled specialists of the cooperating nations have called for the development of a Unified System of Electronic Computers [YeS EVM] that are compatible in hardware, information support and software.

Today as a result of cooperation, socialist nations have solved practically all problems of developing a powerful industry for producing computer facilities. About 30 scientific research institutes and design offices and more than 70 plants are working on agreed plans. A large number of scientific research, design and planning and technological organizations of various ministries and agencies are involved in developing automated management systems.

In 1970-1973 six models of the "Ryad-1" series of YeS EVM computers (YeS-1010, Hungary; YeS-1020, USSR and Bulgaria; YeS-1021, Czechoslovakia; YeS-1030, USSR; YeS-1040, East Germany; YeS-1050, USSR) went through the first international tests and were put into series production, along with the peripheral devices and software required for their operation. The range of productivity of the "Ryad-1" computers is from 3,000 to 500,000 operations per second (op/s).

In parallel with organization of series production of the newly developed equipment, it was also improved. The updated computers (YeS-1012, Hungary; YeS-1022, USSR and Bulgaria; YeS-1032, Poland; YeS-1033, USSR) have made a good showing, and are now the basis of computer output in the cooperating nations.

The first "Ryad-2" computers made their appearance in 1978. By the beginning of 1981, international tests had been done on the YeS-1015, Hungary; YeS-1055,

3

East Germany; YeS-1025, Czechoslovakia; YeS-1035, YeS-1045, USSR, and others. These have much greater capabilities and enable expansion of the class of scientific research, economic and production problems that can be solved. Older models of the "Ryad-2" series of YeS EVM computers have productivity of 1-4 million op/s.

In the same year of 1978, the first complexes based on minicomputers made their appearance. At the present time, joint tests have been done on four types of processors of the minicomputer system (SM EVM) -- SM-1P, SM-2P, SM-3P, SM-4P and some peripheral devices.

The development of these systems was prompted by the necessity of providing computer facilities for those sectors of the national economy where the use of large computers is economically or technically inadvisable. By this we mean the automation of production and technological processes, scientific experiments and research, design work, monitoring systems, the areas of commerce and service, automation of work in agriculture, in transportation, in machine tool building. Minicomputers are also widely used in complexes with large general-purpose machines. In this case they act as complex programmable terminals, coupling processors in remote handling and so on. Therefore the development of SM EVM systems is being done in such a way that the component machines are an organic complement to the existing and proposed YeS EVM computers.

The first microcomputers of class SM-50 underwent joint testing in 1979. These are more economical, compact, and reliable in operation. They are realized mainly on a single board, and are intended for use in various sectors of the national economy, measuring instruments, consumer goods and so on, as built-in control systems.

At the present time, participation of nations in development of computer technology within the framework of the Intergovernmental Commission is characteri0ed by the following data (see the table).

	Tested devices in 1970-1980								
	Bulgaria	Hungary	GDR	Cuba	Poland	Rumania	USSR	CSSR	Total
YeS EVM	38	46	23	—	21	2	49	49	228
SM EVM	29	22	23	5	14	4	23	32	152
Total	67	68	46	5	35	6	72	81	380

In 1979, the second international exhibition of facilities developed within the framework of the Intergovernmental Commission was held in Moscow. All YeS EVM computers of the "Ryad-2" series and SM EVM systems (first phase) were demonstrated there together with their use in the national economy of the member nations.

Today in addition to this, concepts have been developed and preliminary drafts have been drawn up for the "Ryad-3" series of YeS EVM computers and the second

phase of the SM EVM system. Experimental design work is being done on the basis of these drafts.

The "Ryad-3" series of YeS EVM computers includes more powerful and economic computer equipment with high efficiency. This equipment will be typified by:

further reduction in cost of hardware, considerable reduction of overall dimensions and power consumption per unit of processed information. This will be achieved by extensive use of large-scale integration (LSI chips) and new methods of designing technological facilities, sequential unification of functional and engineering features, wide use of modular principles, dataway organization of exchange and functional microprogram superstructure;

increased productivity and throughput of the system, hardware (microprogram) realization of a number of software functions, use of the principles of parallel calculations;

considerable increase in reliability of hardware and software;

further development of principles of network (collective) use of computer facilities;

improved effectiveness of formulating and solving problems by development and extensive introduction of programming automation techniques.

The "Ryad-3" series will ensure continuity of YeS EVM facilities, namely:

maintenance of compatibility on the level of machine codes;

capability of using any previously developed item of the YeS EVM computer line and currently operating systems, programming for new hardware;

maintenance of existing methods of data representation on external media, external codes and exchange procedures.

Realization of the "Ryad-3" YeS EVM program will enable development of models with range of productivity from 100,000 to 5,000,000 op/s with a broad array of up-to-date peripheral equipment.

Development of the second phase of the YeS EVM system calls for improvement of technical-economic indices, increased speed, expansion of the product list, development of capabilities of problem orientation, improvement of reliability, improvement of the system for checking and diagnosis, and also reduction of overall dimensions.

The second phase of the SM EVM system is subdivided into five classes:

microcomputers—eight-bit machines on one printed-circuit board intended for permanent wiring into various instruments and devices;

computers that are systems-compatible with the first phase of the SM EVM system;

5

TECHNICAL ECONOMIC INDICATORS
OF YeS EVM COMPUTER SYSTEM

```
 ┌─────────────┐      ┌─────────────┐      ┌─────────────┐
 │  Ryad-1     │      │  Ryad-2     │      │  Ryad-3     │
 │  1970-1973  │      │  1977-1979  │      │  1982-1984  │
 │ Speed: from │ ───> │ Speed: from │ ───> │ Speed: from │
 │3,000 to     │      │12,000 to 4  │      │0.1 to 5     │
 │500,000 op/s │      │million op/s │      │million op/s │
 │Memory volume│      │Memory volume│      │Memory volume│
 │ 64-1025     │      │ 0.2-16      │      │ 2-16 Mbytes │
 │ kbytes      │      │ Mbytes      │      │             │
 │  K = 1      │      │  K = 2.5    │      │  K = 15     │
 └─────────────┘      └─────────────┘      └─────────────┘
```
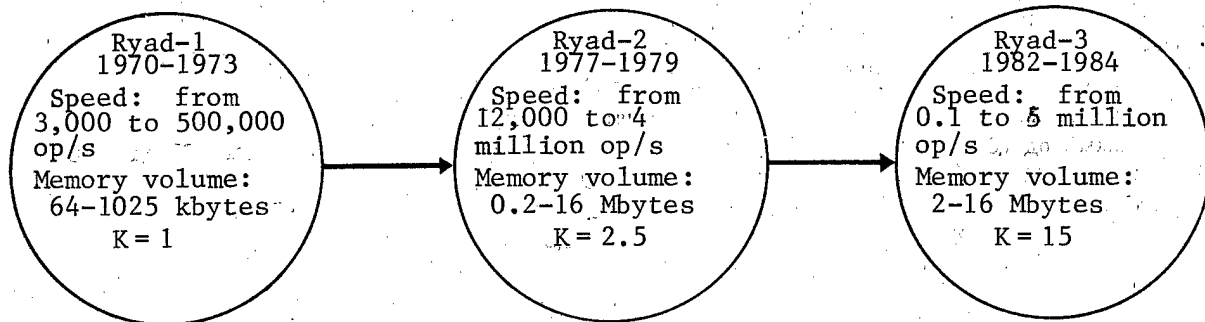
K--Relative productivity/cost factor

models of high-productivity computers. Will be used in real-time systems.
Compatible with younger SM EVM models. Have wider capabilities for real-time
operation. Ensure operation with virtual memory volume of up to 16 Mbytes;

special processors;

multiprocessor and multicomputer complexes. Constructed on the basis of models
of preceding classes and special processors.

Second-phase SM EVM systems will ensure:

on-line data processing;

creation of complex automated control systems that require a large volume of
centralized data processing. When requirements are met, will have elevated
reliability and systems viability;

creation of distributed data processing systems;

high productivity of processing large volumes of information with capability
of using parallel calculations in the algorithm;

The microelectronic element base is decisive in raising technical-economic
indices of the computer facilities being developed. The Intergovernmental
Commission has given particular attention to advanced development and production
of the microelectronic element base for the "Ryad-3" series of YeS EVM computers
and the second phase of the SM EVM system.

As a result of considerable cooperative work of participating nations, the
volume of production of computer hardware over the last ten years has increased
by a factor of more than 6, while mutual shipments among nations have increased
by more than 23-fold. Because of this, today the nations of socialist

Increase in production volume and commodity turnover of computer goods in nations that are parties to the Agreement (1971-81)
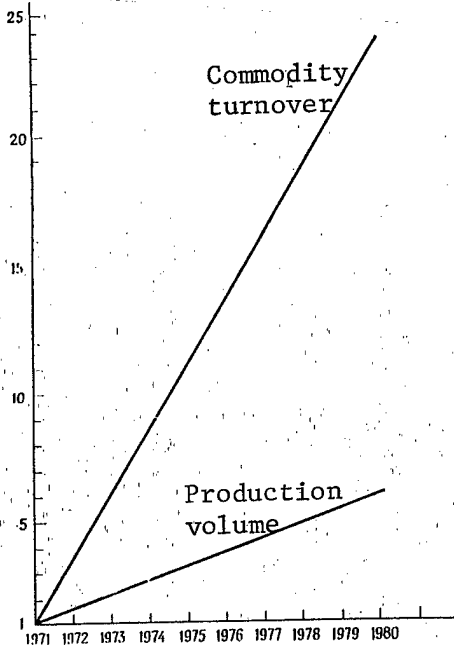
Volume of YeS EVM and SM EVM hardware developed, produced and used in nations that are parties to the Agreement
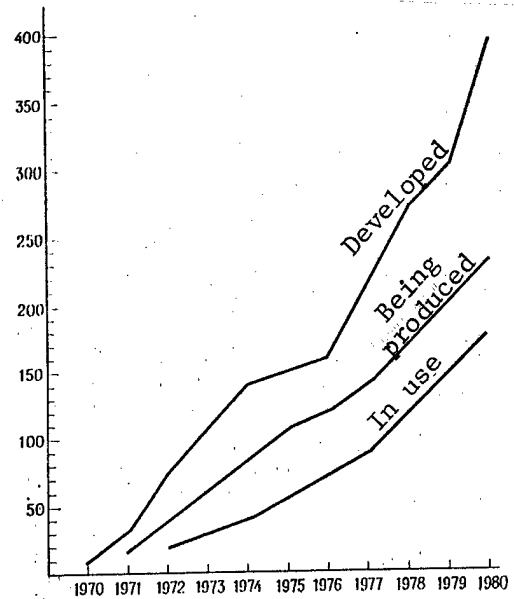
friendship are meeting their needs for up-to-date computer technology by their own production and by shipments from other brotherly nations.

Extensive international division of labor in the development and production of computers has been effective in the highest degree. Calculations show that concentration of production as a basis for extending series production by a factor of 3-4 reduces the cost of goods produced by 35-40% and increases labor productivity by a factor of 2-2.5

Cooperation in this area is now entering a new phase. Based on the necessity of attaining a higher technical level of computers and raising the efficiency of producing and using them, and also for the purpose of more complete satisfaction of needs, the government leaders of CEMA member nations at the Thirty-Fourth Meeting of the Council Session signed an Agreement on Multilateral International Specialization and Integration in Developing and Producing Computer Equipment.

The agreement stresses comprehensive resolution of problems. This means that cooperation will be developed from scientific research and design work to mutual shipments. Provisions are made for attaching to each nation a certain "section" in the development and manufacture of computer equipment.

The agreement establishes six functional groups of computer equipment in which multilateral division of labor is implemented: "Computers and Processors",

"External Storage Devices", "Data Input/Output Devices", "Remote Data Processing Devices", "Data Preparation Devices", "Subassemblies, Modules and Components".

The agreement defines major tasks of the nations for the long-range future. Even during preparation of the agreement, the optimum product list of computer facilities was outlined, steps were worked out for updating the list, and also for ensuring high effectiveness of utilization. To attain a high technical level, provisions have been made for systematic examination of such core problems as advanced development of the microelement base, continuous increase in the technical-economic indicators of devices under development, deepening of the division of labor through transition from item to subassembly and component specialization. There are big problems to be solved in the area of unification and standardization of individual components and subassemblies as well. This will ensure maximum utilization of developed production capabilities and further improvement of cooperation in this area.

The attachment of definite groups of items to the nations in doing research work leads to more organized turnover of research starts. This will raise the general level of the scientific potential necessary to attain high quality of computer facilities being developed, bringing them into line with world standards.

Specialization will enable more goal-directed involvement of academies of sciences and other scientific institutions in resolving the major issues in this area. Systematic analysis of work in progress will be conducive to making sound long-range decisions in defining the concepts for further development of computer technology.

Specialization also fosters improved relations among organizations of the nations responsible for development, production and sales of computer facilities. It plays an important part in raising the technical level of the items being developed, their competitiveness, providing for exports to outside nations. It is also of great significance in more rational utilization of scientific and design personnel in production capacities, and organization of larger-series production, improvement of quality and reduction of cost of items.

The agreement provides in the current five-year plan for volumes of mutual shipments of up-to-date computer facilities in an amount of more than 15 billion rubles, which is approximately double the level for 1976-1980.

Note should be taken of the high level of specialized shipments of computer goods. The fraction of specialized shipments of CEMA member nations in export of all kinds of machines and equipment has reached 35%; this fraction is more than 80% in export shipments of computer facilities.

Within the framework of the new agreement, the latest computer facilities will be developed and put into series production. A considerable improvement is being made in such an economic index as the "productivity/cost" ratio. Developed software, instrumentation and organizational facilities of programming technology will considerably simplify the use of computers. There will also

8

be qualitative changes in the structure of the computer pool. This will maximize satisfaction of the needs of nonprofessional users for computer service.

Resolution of the problems of remote processing will enable more active development of computer networks. These will improve effectiveness of introducing computers in different areas of science, business and production through integrated use of computer capacities and collective access to the resources of the entire network. Setting it up requires solution of many fundamentally new problems. Among these are interaction of hardware and software, economic and legal regulation of operation of the network under conditions of wide access to its resources, including on an international scale. Also included here is an estimate of information flows to be transmitted and processed within a nation and between nations, organizational problems of putting the networks into operation, and also economic advantages of using them, working out agreeable concepts of construction.

Rapid development of such computer networks is anticipated for the early eighties. Unification of local and large national and international networks will continue. New principles are being developed for organizing computer networks, and for processing data that have already begun to be realized in regional and sectoral management systems. Primary among these principles is distributed data processing. Most important among them in the area of software on the new stage of cooperation are:

expansion of base software;

producing advanced equipment for developing applied programs;

extensive use of computer facilities for automating the programming process.

Comprehensive servicing of computer facilities is very important at the present stage. A distinguishing feature of this technology is that work results reflect the results of cooperation of socialist nations in the area of development and production, as well as the use of computer facilities.

Taking consideration of the accumulated experience in developing a powerful sector of computer technology within the framework of socialist cooperation, we must give more attention to problems of joint planning of computer production and more effective utilization of this equipment, a more comprehensive approach to solution of problems that arise.

The division of labor and integration of efforts of socialist nations is a basis for solving problems now facing the brotherly nations. Therefore, as pointed out by Comrade L. I. Brezhnev at the Twenty-Sixth CPSU Congress, our party and the other brotherly parties "are steering a course toward transformation of the forthcoming two five-year plans into a period of intense productive and scientific-technical integration of the nations of socialism."

In following this course, the CEMA member nations will reach new successes in building socialism and communism.

CEMA AGREEMENT ON DEVELOPMENT IN MICROELECTRONICS

[Article by Mikhail Rakovskiy, director of Coordination Center, Intergovernmental Commission on Cooperation of Socialist Nations in the Field of Computer Technology: "New Stage in Development of Microelectronics for Computer Facilities"]

[Text] At the Thirty-Sixth Meeting of the CEMA Session, considerable attention was given to new priority areas of specialization and integration of production in the field of machine building, and in particular in microprocessor technology and microelectronics.

Comrade N. A. Tikhonov, chairman of the USSR Council of Ministers, noted in an address to the Session that concentration of efforts on major sections of scientific-technical progress, producing stockpiles for making equipment that saves energy, materials and labor, as well as facilities for automation and mechanization based on the latest advances in electronics are matters of primary importance under present-day conditions.

Practice shows that further advances depend on increasingly closer cooperation among brotherly nations in areas of high technology that revolutionize all social production. It is these areas that today ensure not only an increase in labor productivity, but also a change in its social conditions.

The Agreement on Multilateral International Specialization in Developing and Producing Items of the Microelectronic Component Base (MCB) signed at the

10

Thirty-Sixth Meeting of the CEMA Session in application to computer technology, special technological equipment and ultrapure materials for microelectronics is expected to be conducive to practical implementation of the work. Its purpose, on the basis of deepening international socialist division of labor, is the most complete satisfaction of the needs of participating nations for these commodities, and expanded industrial production of these goods. The Agreement includes Main Directions and Obligations of Signatories.

It defines:

specializing nations responsible for development and for ensuring conformity of developments of specialized MCB items as well as items of microelectronics to world attainments, continuous improvement of technical characteristics and technology of manufacture, series production and competitiveness of goods on the world market;

coordinator nations that handle coordination of developments and production of MCB items and items for microelectronics by specializing nations in accordance with the Main Directions through their own corresponding organizations;

other participating nations that assist these nations in attainment of high technical-economic parameters of specialized items by transmitting corresponding experience on using them in their own nations.

In defining the specializing nations, consideration was taken of traditional areas of activity of their combines and enterprises having experience in the area of:

computerized design systems--Bulgaria and the USSR;

computerized measurement equipment--Hungary;

opticomechanical equipment--East Germany;

equipment for assembling integrated microcircuits--Poland;

equipment for processing semiconductor materials--Rumania;

equipment for making structures of large-scale integration (LSI) and ultralarge-scale integration (ULSI) chips--USSR;

analytical and monitoring-measurement equipment--CSSR, East Germany and so on.

The Main Directions of division of labor and unification of efforts of participating nations were prepared by leading specialists of their electronics sectors.

This agreement is a development of the framework General Agreement on Multilateral Cooperation in the Area of Development of a Unified Standardized Base for items of electronic technology, special technological equipment, semiconductor and special materials for producing them signed at the Thirty-Fifth Meeting of the CEMA Session.

11

The Intergovernmental Commission on Cooperation of Socialist Nations in the Area of Computer Technology within the framework of the new agreement will handle coordination in development of items of the microelectronic component base for all kinds of computer facilities. This includes single-chip and single-board processors, microprocessor series, large and ultrafast computers, and a variety of peripheral equipment.

The fact is that at the present time further success in developing and using computer hardware is most intimately tied up with results of scientific research in the area of microelectronics and development of production of MCB items based on the latest special technological equipment and ultrapure materials.

The latest technological principles of creating microelectronic devices make revolutionary changes in development of computer equipment. This shows up with particular clarity in small and single-board computers.

In developing single- and multichip microprocessors, and single-board microcomputers, the problem is to produce inexpensive and reliable hardware that handles all forms of regulating and controlling factors that act on the process. Instrument makers are to play a large part in solving this problem.

At the present time, scientists and production workers of socialist nations must generalize the entire theoretical stockpile on automatic processing of digital information for control of many mechanisms and machines, including electric drives. This is because without automatically controlled electric drives, expansion of the use of microprocessor technology will be limited.

Becoming increasingly important today is acceleration of development and series production of promising microprocessor components based on LSI chips, which requires development of new methods of design and improved technological processes. This necessitates careful coordination and interaction of scientific research and experimental design work in the area of microelectronics and work in the area of computer technology.

Use of the latest microelectronic component base produced by socialist nations and standardized for individual functional groups of computer facilities is a major condition for carrying out a unified technical policy in developing different forms of equipment.

To support this policy in development of computer facilities and coordinated development among socialist nations in the production of MCB items and items for microelectronics aimed at complete satisfaction of needs through domestic production, specialization must be intensified in the shortest possible time, with provisions for an increase in capital investments in appropriate sectors of the national economy. Particular attention should be given to microprocessor LSI chips and memory LSI chips.

Naturally, the development of the microelectronics component base requires development of the latest special technological equipment and ultrapure materials. The coordinated efforts of participating nations in the agreement are being directed toward solution of this problem.

To increase efficiency of producing MCB items, the specialists of socialist nations have selected and coordinated a standard progressive technological process for making LSI and ULSI chips and promising models of special technological equipment for the following integrated sections:

producing plates of semiconductor materials;

producing LSI and ULSI structures on semiconductor plates;

assembling integrated microcircuits into chassis of various types;

checking parameters and testing LSI and ULSI chips;

standardized system for designing LSI and ULSI chips;

production of various phototemplates, chassis for LSI and ULSI chips, cleaning and checking media and so on.

In choosing promising models of special technological equipment, consideration was taken of the best achievements of socialist and other nations.

The complex of special technological equipment includes a large number of analytical and measurement-monitoring equipment enabling verification of reliability of the process of making LSI and ULSI chips. This equipment is conventionally subdivided into three groups (cooresponding to three levels of technical specifications). They are intended for producing:

ordinary integrated and high-speed microcircuits of mass series;

LSI microprocessors, LSI memories and single-chip 16-bit and 32-bit microprocessors;

ultralarge-scale integrated microcircuits and ultrafast integrated microcircuits including matrix LSI chips.

In determining the makeup of a complex technological line, provisions are made for hardwiring separate units of equipment in accordance with the approved modular-size network with established placement of facilities for manipulation, conveying and cassetizing, and observance of ergonomic requirements and technical esthetics.

Sections of the complex are controlled by automated systems based on microcomputers, while complex units of special technological equipment are additionally controlled on the basis of microprocessors. These include devices for epitaxial crystal growth, diffusion systems, ion doping, electron beam installations, facilities for registration and projection exposure, systems for measuring LSI and ULSI parameters, and so on.

Primary attention is given to developing production of promising technological equipment, reducing labor inputs in making items of microelectronics, and also introduction of new technological processes and methods that improve reliability

of MCB items. The best forces of scientists and production workers of all participating nations have been enlisted to carry out this work, which is of broad coverage.

Particular attention should be given to the mutual tie-in of work on developing MCB items and computer facilities, which is being realized through systems of computerized design.

The standard designing system that is being developed ensures production of LSI and ULSI chips, including matrix chips, phototemplates through image generators, and tie-in with a system for computerized design of modules, standard substitution components, and computer facilities.

The system for computerized design of LSI and ULSI chips is realized on the basis of equipment produced by nations participating in the agreement.

The high complexity of the new generation of special technological equipment for producing LSI and ULSI chips and the need for ultrapure materials have necessitated an increase in the capital investments in the sector and expenditures on research and development to come up with new technological processes, equipment, materials.

The attention being given to development of microelectronics can be attributed to the infinite capabilities that this field creates for increasing the capital-output ratio, raising labor productivity and resolving many major social issues. Chief among these is freeing man from routine and heavy work.

Socialist nations have a sufficiently powerful scientific research, design and production base to develop and produce items of microelectronics, special technological equipment and ultrapure materials. The work being done within the framework of the agreement will expand each year, reflecting new prospects and realized results of scientific-technical progress in this area.

The mutual interests of all nations that are parties to the agreement is confirmed from year to year by the increasing commodity turnover in items of microelectronics and special technological equipment.

Realization of the agreement will enable participating nations in a relatively short time to achieve a considerable elevation of the technical level and scales of production of items of the microelectronics component base not only for hardware in computer technology, but also for microprocessors, industrial robots and so on. And this will have an effect on raising the technical level and labor productivity in many other sectors of the national economy of CEMA member nations.

6610
CSO: 1863/21

RIGA-BUDAPEST COMPUTER LINK VIA TELEPHONE

Moscow GUDOK in Russian 3 Nov 82 p 4

[TASS report: "Hello, This Is the Computer"]

[Text] Two computers, in Riga and Budapest, had plenty to say to each other over the telephone. An experiment which has been completed lasted for a week. It was controlled from the Institute of Electronics and Computer Technology of the Latvian SSR Academy of Sciences. Despite the noise that occurred, the machines steadily maintained a dialog and communicated without failures. They themselves located and eliminated the errors that occurred in the exchange of test information. Thus, software compatibility was checked out and hardware was tested for interfacing computers with communication channels.

"The experiment was conducted by Soviet and Hungarian scientists who are developing computer networks," said E. Yakubaytis, vice president of the Latvian SSR Academy of Sciences. "They are also being developed in the other socialist community countries as well. This will allow in each of them uniting the entire pool of computers and open broad access to information banks over telephonef teletype, radiorelay and other lines."

8545
CSO: 1863/61

MINSK PRODUCTION ASSOCIATION IMPROVES TAPE UNITS

Moscow EKONOMICHESKAYA GAZETA in Russian No 44, Oct 82 p 19

[Article by D. Pislyakov, engineer, Minsk: "On a Creative Quest"]

[Excerpt] At the Minsk Production Association for Computer Hardware, communist A. Lepushenko has been called one of the best. V. Shershen', chief designer at the enterprise, said he is an "experienced specialist and an excellent innovator." Aleksandr Ivanovich heads a group of designers that are developing a tape unit for the Unified System of Computers. The unit is in series production, but the designers under A. Lepushenko are engaged in improving and modernizing the unit: they are looking for and finding more and more new reserves to enhance quality and the reliability of the unit and to reduce the labor input to manufacture it. Aleksandr Ivanovich has made 15 innovative suggestions: 7 have been incorporated into production and 3 are at the stage of introduction. He developed a new unified block of an automatic unit for computers that replaced three. This made it possible to improve the structural design of the unit as a whole, reduce the amount of wiring and assembly, reduce the labor input for manufacturing the device, and save a considerable number of parts in short supply. This same block was then used by the designers in developing a new model of a tape unit. Annual savings were over 200,000 rubles.

8545
CSO: 1863/61

16

SM-1403 USES ITALIAN PERIPHERALS

Moscow EKONOMICHESKAYA GAZETA in Russian No 50, Dec 82 p 22

["Review of World Economic Facts, Figures and Events: Soviet-Italian Business Ties"]

[Excerpts] In Moscow last week was a large group of Italian businessmen headed by R. Ossol, president of the Italian-Soviet Chamber of Commerce. The delegation included managers of well known Italian firms (ENI, FIAT, Finsider, FATA, Montedison, SNIA-Viskoza, Koye and Klerichi and others). The Italian delegation was welcomed by I. V. Arkhipov, first deputy chairman, USSR Council of Ministers; N. S. Patolichev, minister of foreign trade; N. D. Komarov, chairman of the Soviet part of the Commission on Economic, Scientific and Technical Cooperation between the USSR and Italy and first deputy minister of foreign trade; D. M. Gvishiani, deputy chairman of the State Committee for Science and Technology; and other officials from Soviet ministries and departments.

Soviet specialists were briefed on the export program of the 14 firms making up the Italian Olivetti concern. The program is broad in scope, ranging from a huge processing center to a miniature electronic calculator.

"We appreciate highly the results of cooperation with Soviet organizations," Franco de Benedetti, vice president of the concern, noted at the press conference. "We are looking for new points of contact of interests. And it is not just a question of mutual export or import. Complexes based on the Soviet SM-1403 computer with Italian peripherals were demonstrated at the exhibit in "Sokol'niki." This may evoke interest in third countries."

Shown in the photo is an electronic computer complex based on the Soviet SM-1403 computer with an Italian peripheral.

8545
CSO: 1863/61

17
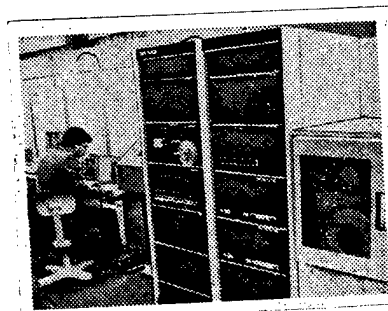
ENGINEER RECEIVES GOLD MEDAL FOR ACCELERATING COMPUTER PRODUCTION

Minsk SOVETSKAYA BELORUSSIYA in Russian 15 Aug 82 p 2

[Article by D. Pislyakov, Engineer]

> [Text] The Main Committee of the Exhibition of Achievements
> of the National Economy of the USSR has awarded a Gold Medal
> to L. V. Chaadayeva, designing engineer of the Minsk Order
> of Lenin Plant imeni G. K. Ordzhonikidze, for her great
> contribution to mastering output of a new powerful elec-
> tronic computer.

At Minsk Order of Lenin Plant imeni G. K. Ordzhonikidze, startup of series pro-
duction of the YeS-1060 computer is well remembered. It was a hectic time.
The best specialists had been working on development of design documentation
and perfection of the technological process. Painstakingly working along with
others was First-Class Designing Engineer L. V. Chaadayeva. Since 1959, Lyudmila
Vladimirovna has been working at the plant in the design office. She has taken
an active part in mastery and production of many Minsk Computers.

The first YeS-1060 computer had been produced by the set deadline. And designer
Chaadayeva was pleased with the success. At the same time, she knew full well
that there was a lot remaining to be done to improve the quality indicators
of the new machine.

Lyudmila Vladimirovna kept coming back to the electronic circuitry of the logic
plug-in cards, and came to the conclusion that many decoupling supply capacitors
were being inefficiently used. Wouldn't it be possible to eliminate some of
them and turn their functions over to capacitors in the resistor-capacitor
modules? She shared her thoughts with those in charge of the design department.
Her idea was approved and supported.

A little later, the rationalization suggestion had been accepted and put into
production. This cut down labor inputs on making the logic cards, and saved
a considerable number of scarce components. The annual savings from introducing
the innovation has been 100,208 rubles. The experience of Minsk computer de-
velopers has been displayed at the main exhibition of the nation.

Lyudmila Vladimirovna is now chief of the design office. Working in the collec-
tive energetically and with creative enthusiasm are such experienced designing

engineers as Alevtina Sergeyevna Konyuchenko, Mikhail Davidovich Rozenshteyn, Vladimir Vladimirovich Shumskiy, and also young specialists Yekaterina Bel'skaya, Ivan Gayduk, Leonid Rodyno and others. In a word, they are people of like mind. The office holds a leading place in socialist competition not only in the department, but in the plant as well.

Chaadayeva remains an example to others as before. Not wanting to rest on her laurels, as the saying goes, Lyudmila Vladimirovna has developed and submitted another rationalizer suggestion aimed at improving the control panel of the "sixty" processor. Work is now in progress on putting it into production. This will save many thousands of rubles more of State money.

6610
CSO: 1863/20

NEVA-1M CONTROL COMPLEX FOR COMMUNICATION SWITCHING SYSTEMS

[Article by Anatoliy Grigor'yevich Kukharchuk, candidate of engineering science,
SKB MMS IK AN USSR [Special Design Bureau for Mathematical Machines and Systems,
Institute of Cybernetics, UkSSR Academy of Sciences] (Kiev); Andrey Ivanovich
Nikitin, doctor of engineering science, UkSSR Academy of Sciences Institute of
Cybernetics (Kiev); and Leontiy Antonovich Strutinskiy, SKB MMS IK AN USSR (Kiev)]

[Text] Introduction. In the last 10-15 years in communication switching systems,
particularly at stations and nodes for switching telephone channels and lines, more
and more use has been made of electronics and computer hardware. The electro-
mechanical switching systems in operation (ten-step, crossbar) have reached the
limit of their capabilities with respect to technology (speed, reliability) and
functional capabilities. Also, the cost of manufacturing and maintaining them has
reached a stable level that will be difficult to improve. Replacing them are the
quasi-electronic and electronic systems which will be used as the base in the long-
term to establish a single integrated network with digital systems for transmission
and switching equally suitable for sending voice, imagery and data [1-3].

Quasi-electronic switching systems are stations and centers with space-shared chan-
nels, in which the switching field is built on the basis of hermetically sealed
reed relays, crossbar switches and other miniaturized, high-speed, electromechani-
cal elements. In electronic systems, the switching field is built with electronic
elements and, as a rule, time-shared channels with pulse-code modulation are used.

By type of control, electronic switching systems (from here on, the term "electron-
ic" includes quasi-electronic systems too) are divided into systems with hard-wired
logic, programmed logic (program is stored in a ROM) and systems with a program
stored in main memory (i.e., systems with program control). The most advanced are
the latter systems in which specialized computers are used as control units. The
main advantages of these systems are:
--considerable reduction in size of equipment, increase in reliability and, thus,
reduction in costs for installation and maintenance;
--simpler and more economical interfaces with existing stations and centers; and
--savings for cable networks (for digital systems with time-shared channels).

Also, using program control permits automating the most laborious item in operating
the stations and centers: maintenance, which in turn permits reducing service

personnel and operating expenses considerably. Program control enables a high level of flexibility since it allows introducing new service programs without changing equipment (this is especially important when the network configuration is changed) and offering subscribers services that would be impossible without program control (less dialing for subscribers called most often, automatic rerouting of calls to another telephone, call waiting when the number dialed is busy, conference calling of three or more subscribers and others).

Research in electronic switching began at the start of the sixties, but just by the end of the sixties, these systems began competing successfully with the electromechanical.

The most extensive scientific research and practical development in electronic switching is performed in the United States, Japan, France, Sweden, the FRG, Canada, Belgium and Holland [4].

A number of systems are at the stage of being placed in trial operation in the USSR; this includes the quasi-electronic, automatic, long-distance telephone exchange with the Neva-1M control complex (UK).

The Neva-1M control complex was developed at the UkSSR Academy of Sciences Institute of Cybernetics imeni V. M. Glushkov in cooperation with Robotron (GDR) where the system is in series production.*

The Neva-1M is designed for use in communication switching systems for various purposes with medium or large capacity, and in particular in:
--quasi-electronic local and long-distance automatic telephone stations and centers;
--electronic automatic telephone stations and centers in an integrated digital communication system; and
--message switching centers.

A number of requirements have been imposed on the computers used as central controllers for communication switching systems; the main ones are:
--high degree of reliability during lengthy periods of operation;
--real-time operation;
--high throughput;
--general-purposeness in the sense of the capability of introducing new functions, and handling tariffing, acquisition of statistics and others in addition to control.

These and some other requirements have had a major impact on the architecture and logic structure of the computers making up the complex and have determined the set of operations and other characteristics [5].

Architecture and Main Technical Parameters. In foreign communication switching systems, reliability is ensured, as a rule, by controller redundancy. Following

---

the American ESS1 system, the majority of foreign systems use redundancy through synchronous operation of machines. The continual comparison of information coming into each machine allows essentially instantaneous detection of any failure. Hardware and software facilities enable detecting the computer that has malfunctioned and preventing it from affecting system operation.

In the Neva-1M control complex, a duplex computer system made up of two specialized control computers (SUM), hardware redundancy is used at the level of the individual functionally independent devices: central processor (TsPR) [CPU], peripheral processor (PPR), I/O channels (KVV), main memory (OZU) and the interface unit (USZU) and service console (figure).

Key:
1. Switching system peripherals
2. Neva-1M control complex external units
3. Neva-1M control complex
4. First specialized control computer: [units, reading left to right]
      service console
      CPU
      peripheral processor
      I/O channels
      interface unit
      main memory
5. Second specialized control computer [with same units]

The CPU is intended for program processing of information in the process of implementing functions to control the switch system in real time and the functions to operate the system. The peripheral processor organizes exchange of information with peripherals in the switching system and enables preprocessing of information coming from these units. The I/O channels, one multiplexer and two selector, handle exchange of information between the complex and external units in the Neva-1M control complex.

The CPU, peripheral processor and I/O channels, which are, respectively, general-purpose and specialized processors, have direct access to main memory (through the interface unit) intended for storing programs and data. Part of main memory, operating in the read-only mode, is used to store resident programs, constants and data rarely changed. The other part, operating in main memory mode, enables

22
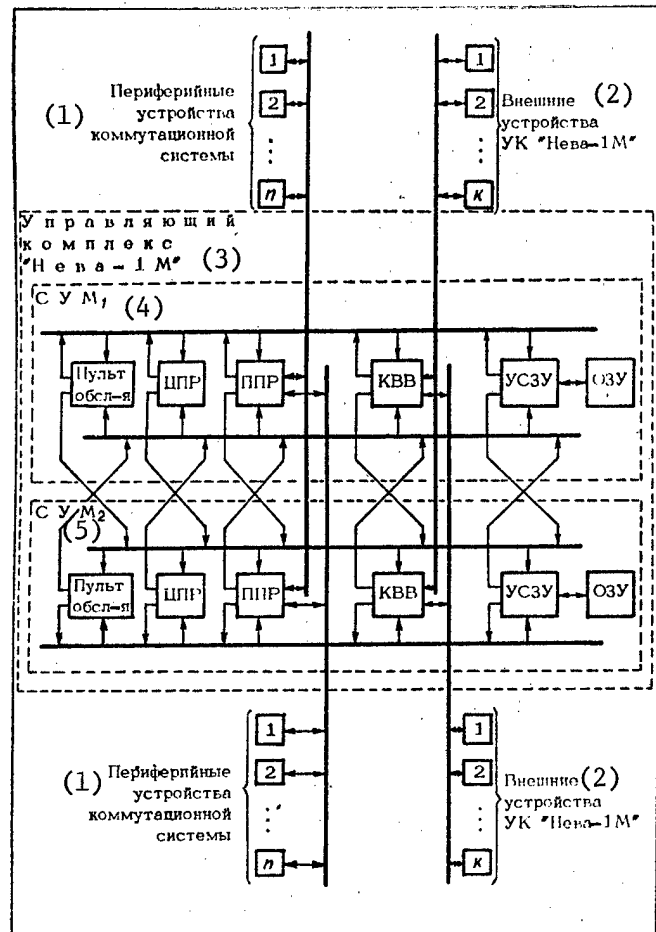
storing working information (mapping files, working fields of programs and others) and programs and data for background tasks called from external storage (programs and data for tariffing, acquisition of statistics, diagnostics of devices that have malfunctioned and others).

The console is used to implement complex servicing functions by personnel both in the normal complex operating mode and when a particular device is out of order.

All functionally independent units in the complex are interfaced by two systems of standard buses: the system of buses joining devices in "their own" machine, and a system of buses joining devices in "their own" machine with those in the "other" machine. Information and address buses in these bus systems are unibuses while control buses are radial.

The complex-external devices interface conforms to the interface for the Unified System of Computers. All external devices belong to the nomenclature of the Unified System of Computers (printers and perforated tape devices, magnetic tapes, magnetic disks, displays and others); the set of them is governed by the requirements for the specific switching system.

Structurally, the complex consists of three racks. Each specialized control machine in the complex is made in the form of a standard, separate, three-bay rack (Unified Computer System design); one houses the CPU, main memory and interface; the second houses the I/O channels and logic for the service console; and the third houses the peripheral processor. Between the specialized control machine racks is the distributing rack that houses the flat cables with intermachine connections, connectors for connecting switching system peripherals and panels for servicing of the specialized control machines.

All devices in the complex are supplied by individual sources (converters) that generate the voltage needed for each device. The primary power supply for the devices in the complex, which is fed to input for the converters, is a guaranteed station voltage of 60 V.

In the normal mode of operation, the machines in the complex operate synchronously and compare the information coming from pairs of like devices at the input to each of the devices. Synchronous operation of the machines is supported by clock pulses from one machine going to all devices in the complex. In the process, there is the capability of switching units manually or by software to operation from the clock in the other machine.

If any device in the complex malfunctions, which is detected by the comparison circuits or (and) the internal checking circuits, this device is removed from the operating configuration (by software or hardware facilities) and the complex continues normal functioning, but with incomplete backup. Only when two like devices malfunction is the complex made inoperative as a whole.

After any device is removed from the working configuration, the complex is automatically switched to the mode of control of the switching system with diagnostics (at a low priority) for the device that has failed. Using specialized hardware facilities that provide diagnostic routines direct access to the apparatus of

devices both after execution of a particular operation and after execution of the next step in an operation allows diagnosing a device that has failed with a precision to one or more standard cards.

To diagnose main memory, a special apparatus is used that allows making test checks at the maximum rate of access to main memory, i.e. under the most intensive conditions for this unit which cannot be implemented by software. This same test apparatus is used in case of malfunctions and also in periodic preventive maintenance of memory when elements with limit values of particular parameters have to be detected.

The Neva-1M control complex affords control of a quasi-electronic, automatic, long-distance telephone station with a capacity to 10,000 X 2 trunks with a maximum load of 0.7-0.8 erlangs per line. The equipment in the complex is designed for round-the-clock operation under stationary conditions.

Main Technical Parameters for the Neva-1M Control Complex

Average speed of CPU (for switching system control tasks)	750,000 operations/sec.
Maximum size of main memory	1M bytes
Channel throughput: multiplexer	50K bytes/sec.
selector	1M bytes/sec.
Set of external devices	nomenclature of Unified System of Computers
Interface with external devices	Unified System interface
Design base	Unified System designs
Element base	KMYe-10 integrated circuits (made in GDR)
Estimated down time of complex (when two like devices malfunction)	2 hours in 20 years of operation

Logic Structure. In its structure, each specialized control machine in the complex is a multiprocessor computer with a common extent of memory. The CPU, peripheral processor and I/O channels operate simultaneously and each of these units has direct access to main memory (through the interface unit). Also, the CPU has direct access through the direct control channel to all devices in the complex to put them into operation (in the normal operating mode) and for diagnostics (during a malfunction).

The multiprocessor principle of design of the machines in the complex in solving the tasks of control of switching systems affords simultaneous implementation of the following functions:
--exchange of information with peripherals in the switching system and preprocessing of it;
--information exchange with external devices in the Neva-1M control complex; and

--processing of information that has arrived from peripherals and external devices in accordance with the specified discipline service.

As already noted, the first two functions are implemented by the peripheral processor and I/O channels, respectively, and the third function, by the CPU. Thus, in the quasi-electronic, automatic, long-distance telephone station, the peripheral processor implements the functions of scanning trunk sets, generating numbers of sets through which the successive digits of dialing or interaction signals have come in, issuing control effects on switching elements in the system in accordance with intermediate paths found in the switching field, etc.; the I/O channels implement the functions of information exchange with external devices in the control complex in sending messages to an operator, loading main memory with background tasks, etc.

The peripheral processor and I/O channels are narrowly specialized devices with programmed logic, the operation of which is initiated by the central processor by issuing the appropriate commands over the direct control channel. Messages on completion of work by the peripheral processor and I/O channels go to the CPU through the interrupt system. Implementation of simple and frequently repeated functions in solving problems of switching system control by the specialized devices allows reducing CPU load 40 to 50 percent and thereby almost doubling complex throughput.

The CPU is a general-purpose processor with program control. The CPU instruction set and logic structure are oriented to solving the problems of controlling communication switching systems. The CPU instruction set has a total of 144 instructions (with regard to modifications) and includes in addition to the universal set of operations (addition, subtraction, multiplication, bit-by-bit logic operations, conditional and unconditional branches, and others), operations that are typical for problems of controlling switching systems (search for left unit, bit recording, bit checking, adding a unit to the code, multiplication of code and others). The instruction set for the Neva-1M control complex also has a number of complex group operations: associative search, search for left unit in an array, search for left nonmatch of two arrays, writing a request to arrays of a queue for a program and others.

The main data formats are the double word (64 bits), the word (32), the halfword (16), the byte (8), the halfbyte (4) and the word with an arbitrary length (from 1 to 32 bits). Arithmetic operations are performed on integers.

The main operations have the following modifications: register-register, register-storage, and register-immediate operand. Memory is accessed by using immediate, direct, relative and indirect addressing of operands.

In the CPU, there are two groups of registers accessible by software: four 32-bit operation-index registers and four base registers. Each operation-index register can be divided into several subregisters (4-, 8- or 16-bit) that participate in operations as independent operation or index registers. Format of operation-index registers (division into subregisters) is specified when a program is initiated and can be changed by special instructions.

25

Operation-index registers are used in programs as scratch-pad storage to speed up computations and to access files by computer numbers of elements which correspond to numbers of channels, lines, peripherals etc. The same registers in some cases can be used as operational, and in others as index.

There are two types of data files (information modules) in the Neva-1M control complex memory: homogeneous and nonhomogeneous. Homogeneous files contain elements of equal length and, as a rule, identical structure. Nonhomogeneous files may have elements with different lengths. All basic information files for switching systems (mapping files, data files and others) are, as a rule, homogeneous. Nonhomogeneous files include, for example, working fields of programs.

Base registers are used to access files during execution of instructions (accesses to the initial area of memory are an exception). Each base register contains the file base (starting address) and information on the file size, type and (in the case of a homogeneous file) the length of file elements. The presence of information on file element length in the base registers allows access to homogeneous files directly by element numbers without advance computation of their addresses. Nonhomogeneous files are accessed by relative addresses (displacements). The presence of information on file size in the base registers allows effecting memory protection when files exceed the limits.

Program modules are specified similarly. The CPU has an instruction address register which holds the base of the program module, its size and mathematical address of the instruction (address of the instruction within the bounds of the module). The program module base and size are entered in the instruction address register when a given program is initiated and subsequently are not changed during its execution. Programs contain only relative addresses both during access for operands (except addresses of the initial area of memory) and during conditional and unconditional branches. The absolute (physical) addresses of the start of program and information modules are unknown to programmers and are loaded to the instruction address and base registers from the corresponding descriptor tables automatically when programs are initiated and when branches to subroutines are made.

The modular organization of the software implemented in the Neva-1M control complex by hardware facilities allows independent compilation of individual programs and prevents a program affecting another when changes are made; it also allows relocating both information and program modules to machine memory without any change to the programs themselves.

Computing Process Organization. Information processing in switching systems, particularly in systems for switching telephone channels and lines, in implementing the algorithms for servicing calls and establishing connections for each call is divided into a series of stages (phases) in accordance with the successive arrival of data from subscribers in a scale of time ranging from fractions of a second to several tenths of a second (lifting the receiver, dialing the number, etc.). From the viewpoint of program processing, each stage is a separate task that has to be handled with regard to the data that has arrived and completed before the arrival of new data, or within the limits of a given time interval. Therefore, such processing involves simultaneous handling of a very large number (tens of thousands) of tasks in the time-sharing mode and in real time.

The efficiency of program control of a switching system to a considerable extent is governed by the efficiency of the facilities controlling the flow of tasks in a machine or, in other words, organizing the interaction of programs in implementing algorithms for call servicing and establishing connections.

The basis of these facilities in the Neva-1M control complex is formed by the program switching system, similar to that which was used for the first time for these purposes in the Dnepr-2 control machine [6], and the apparatus of cyclic files.

Each program in the Neva-1M control complex (except subroutines) is reflected in the program switching system in the form of a so-called interrupt signal stored in a specially organized interrupt signal storage file. All programs and, accordingly, the interrupt signals are distributed by levels of absolute priority of which there are up to 16 (specified when the system is initially set up). The maximum number of interrupt signals is 2,048.

The queue for programs is implemented by using cyclic files in which requests for service on the part of a program corresponding to a given file are stored. To access cyclic files, speacial instructions are used to write to these files and to read from them (only the file number is specified in the instruction). When the instruction is executed by the processor, access is effected to the descriptor table line, corresponding to the given file; this table describes cyclic files and is kept in a fixed area in memory. The line specifies the beginning address of the file, its size and dimension of the elements, which enables immediate access to the file for writing or reading of the next request.

Upon arrival of the first request for service for a particular program, it becomes active since an interrupt signal corresponding to the given program is activated in the program switching system. But actually, the active program with the highest priority is initiated. In the process, if the program belongs to the same (or lower) level as that of the program in execution at the given time, it will be initiated after completion of execution of this program. But when the program level is higher, it will interrupt the program in execution at that time. In the process, the parameters of the interrupted task (and their restoration subsequently) are stored automatically.

When there are several active programs at a given level of absolute priority, the program with the highest relative priority will be initiated. The program of a given level that is interrupted in its time automatically receives the highest relative priority.

Interrupt signals can be masked, and thereby the priority of various programs can be changed by software. Masking is also used to organize special disciplines and to initiate programs, for example, for a temporary schedule. Also, active programs can be transferred to the inactive category (eliminated from the program switching system) by using cancellation by a special instruction of the interrupt signals corresponding to them.

All Neva-1M control complex programs represented in the program switching system are divided into those with or without a queue. Accordingly, interrupt signals too are divided into multi and single call signals. In the process, some programs

27

with a queue can be represented in the program switching system by single call signals:

Programs that in the program switching system correspond to single call signals are initiated one time, i.e. after initiation of this program, it is eliminated from the system (the corresponding interrupt signal is cancelled). After initiation of a program that corresponds to a multi call signal, it is not eliminated from the program switching system (these programs are eliminated from the system only after the queue of requests for them is depleted).

After switching of programs in the Neva-1M control complex, hardware facilities are used for the initial loading of the instruction address register and base registers, and for the initial division of operating registers into subregisters when the programs are initiated. Each program at the beginning has a so-called header containing pointers for loading the base registers (numbers of information files) and a code governing the division of the operating registers. The base registers are loaded with the contents of the corresponding descriptor table (the number in the "header" defines the line number in the table). The instruction address register is loaded with the contents of the other table, the line number of which corresponds to the number of the program stored in the program switching system.

Hardware implementation of some functions of dispatching, specialization of the instruction set to fit switching system control tasks, selection of the data format in accordance with the information formats encountered most often in control tasks, the capability of immediate processing of variable length data and other structural solutions ensure high throughput for the Neva-1M control complex.

Software. The software for a communication switching system built on the base of the Neva-1M control complex depends on system type and purpose and on the types of service implemented in the system. The base part of the software consists of the programs in the base operating system and software facilities in the system for ensuring reliability.

The base operating system is intended for organizing the computing process at all stages of it, including programming, debugging and execution proper of programs in real time. It consists of: the supervisor, data management system, monitor for background tasks, programming system, system for debugging tasks in real time, a set of service routines and generation facilities that allow generating an operating system from the set of facilities for a specific communication switching system.

The main functions of the base operating system are:
--handling the main functional tasks in real time, operator communication and logging the computing process;
--managing data sets on external storage units;
--handling background tasks in parallel with foreground tasks; and
--affording service personnel facilities for programming and debugging of tasks implemented in the control complex.

28

The main functions of the software facilities for ensuring reliability are:
--ensuring complex operating capability when individual devices in it malfunction;
--and ensuring diagnostics of a faulty device and localization of the malfunction.

These software facilities include the supervisor, functional tests for complex devices and diagnostic routines.

BIBLIOGRAPHY

1. Ivanova, O. N., ed., "Avtomaticheskiye sistemy kommunikatsii" [Automatic Communication Systems], Moscow, Svyaz', 1978, 624 pages.

2. Lazarev, V. G., "Elektronnaya kommutatsiya i upravleniye v uzlakh svyazi" [Electronic Switching and Control in Communication Centers], Moscow, Svyaz', 1974, 268 pages.

3. Hobbs, M., "Modern Switching Systems in Electronic Communication," Moscow, Svyaz', 1978, 328 pages.

4. Gurvich, Ye. I., "Characteristic Features in the Evolution of Switching Technology," in "Zarubezhnaya tekhnika svyazi. Ser. Telefoniya, telegrafiya, peredacha dannykh" [Foreign Communications Technology. Series: Telephone, Telegraph, Data Transmission], No 9, 1979, pp 1-10.

5. Kukharchuk, A. G.; Nikitin, A. I.; and Strutinskiy, L. A., "Some Paths of Development of Structure of Processors in Digital Control Computers," USiM, No 6, 1976, pp 22-26.

6. Yegipko, V. M. and Kukharchuk, ed., "Upravlyayushchaya vychislitel'naya sistema 'Dnepr-2'" [Dnepr-2 Control Computing System], Kiev, Nauk. dumka, 1972, 260 pages.

8545
CSO: 1863/37

OPTICAL METHOD FOR ENCODING AND PROCESSING INFORMATION

[Article by Ye.G. Stolov, Leningrad]

[Text] In this article we propose a new approach to the problem of the parallel processing of large amounts of data. In this case the encoding is accomplished by representing the information on each object in the form of a luminous point with a certain radiation spectrum. The physicotechnical realization of this method is based on the following capabilities:
a) the production of a flat, luminous area, each point of which has a certain radiation spectrum (using, for example, a light source, a lens, a spectral instrument and a filament bundle);
b) the simultaneous passage of the radiation from each point in the plane of the luminous area through a spectral filter realized (for example) in the form of two lenses with an interference filter between them;
c) the development of points in the luminous area with a certain brightness of luminescence and the addition and subtraction of images (using, for example, structures containing adjacent layers of a photoconductor and an electro-optical material placed between transparent electrodes [1,2]).

The information encoding and processing methods used here are illustrated with an example of the construction of an optical computer for the solution of nonlinear equations.

A block diagram of such a device is shown in Figure 1a. The computer contains light source 1 with a continuous radiation spectrum and input lens 2, which focuses the source's emissions on the input aperture of spectrometer 3. In the plane of the spectrometer's input aperture there lies the flat end face of encoding fiber-optics bundle 4, the other flat end face of which abuts correcting filter 5 and is located in the focal plane of collimating lens 6. Wedge-shaped interference filter 8 [3,4] and the surface of controllable, three-dimensional light modulator 7, which is adjacent to it, are located in the pupil of the optical system formed by collimating lens 6 and focusing lens 9. Radiation receiver 10 is located in the focal plane of the focusing lens. This receiver makes it possible to accomplish simultaneous and independent registration of an optical signal at all points on the working surface and is based (for example) on structures consisting of a photoconductor and an electro-optical material [1,2]. Such a computer makes it possible to solve nonlinear equations of the type
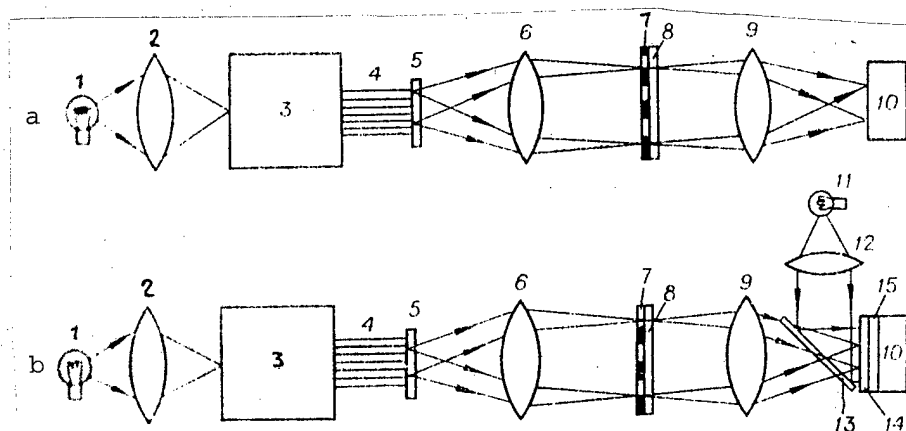
Figure 1. Block diagrams of optical computers for the solution of nonlinear equations.

$$f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) = A, \; x_1 \in \left[x_1^{(1)}, x_1^{(2)}\right] \ldots x_n \in \left[x_n^{(1)}, x_n^{(2)}\right],$$ (1)

as well as systems of equations of the same type.

The computer depicted in Figure 1b contains additional light source 11, and its emissions--after being transformed into a parallel beam by lens 12 and reflected from spectrum separation filter 13--strike the receiver's working surface, in front of which there are a thin layer 14 of a photochromic substance and optical filter 15. This computer makes it possible to find solutions for given intervals of change in unknown nonlinear equations and systems of them.

Without limiting the generality of our discussion, for the sake of simplicity in explaining the functioning of the computer depicted in Figure 1b, let us examine the following nonlinear equation:

$$f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) + \varphi(x_1)\beta(x_2) = B.$$ (2)

We will assume that all functions in equations (1) and (2) are positive.

In order to encode information about x_1, the optical interval of wavelengths $[\lambda_1^{(1)}, \lambda_1^{(2)}]$ is assigned, in connection with which a radiation wavelength λ_1 corresponds to each value of x_1:

$$\lambda_1 = \lambda_1^{(1)} + \frac{x_1 - x_1^{(1)}}{x_1^{(2)} - x_1^{(1)}}\left(\lambda_1^{(2)} - \lambda_1^{(1)}\right).$$ (3)

Analogously, intervals $[\lambda_2^{(1)}, \lambda_2^{(2)}], \ldots, [\lambda_n^{(1)}, \lambda_n^{(2)}]$ are assigned for $x_2, x_3, \ldots, x_n$, and radiation wavelength λ_i will correspond to x_i:

$$\lambda_i = \lambda_i^{(1)} + \frac{x_i - x_i^{(1)}}{x_i^{(2)} - x_i^{(1)}}\left(\lambda_i^{(2)} - \lambda_i^{(1)}\right).$$ (4)

Thus, the luminous "point," the radiation spectrum of which contains n monochromatic lines of equal intensity $\lambda_1 \in [\lambda_1^{(1)}, \lambda_1^{(2)}], \ldots, \lambda_n \in [\lambda_n^{(1)}, \lambda_n^{(2)}]$, represents in encoded form one of the points in the area in which the solution of the equation is sought (the luminous "point" is understood to mean the end face of one of the fibers in bundle 4).
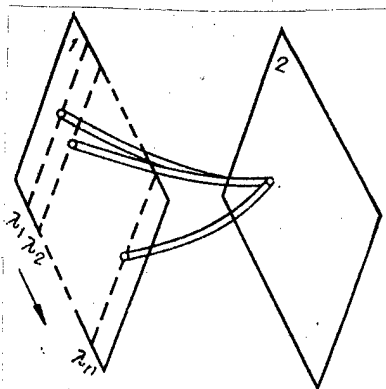
31

Figure 2. Design of a single element of a summing bundle that is part of bundle 4 and represents, in encoded form, one of the points in the area of change of the unknowns. The bundle contains n fibers, in each of which monochromatic radiation on a certain wavelength is propagated: 1. plane of the spectrometer's output aperture; 2. focal plane of lens 6; the arrow indicates the direction of the spectrometer's axis of dispersion.

A continuous spectrum is formed in the plane of the output aperture of spectrometer 3 (see Figure 1a). As a result of the radiation's passage through encoding fiber-optics bundle 4, at the latter's opposite end there forms a luminous area, each point of which represents (in encoded form) one of the points of the area of change in the unknown values; in other words, the spectrum of each point's radiation contains n monochromatic lines belonging to the wavelength intervals $[\lambda_1^{(1)},\lambda_1^{(2)}]$, $[\lambda_2^{(1)},\lambda_2^{(2)}]$,...,$[\lambda_n^{(1)},\lambda_n^{(2)}]$. This effect is achieved through the appropriate choice of the design of encoding bundle 4 (Figure 2). The number of fibers in bundle 4 equals nN, where N is the total number of points in the area of change of the unknowns among which the solution is being sought. In order to represent each point, n elementary fibers are used; their end faces are located in the plane of the spectrometer's output aperture in such a fashion that in one fiber there propagates the monochromatic radiation belonging to wavelength interval $[\lambda_1^{(1)},\lambda_1^{(2)}]$, in another the monochromatic radiation belonging to $[\lambda_2^{(1)},\lambda_2^{(2)}]$ and so on. The opposite ends of these fibers are combined into a single fiber, the end face of which is located in the focal plane of collimating lens 6. Thus, light source 1, lens 2, spectrometer 3 and fiber-optics bundle 4 realize the operation of input into the optical computer of information about all points in the area of change of the unknowns among which the solution is being sought. Having passed through correcting filter 5, the emissions from each point on the end face of encoding bundle 4 are transformed by collimating lens 6 into a parallel beam of rays (correcting filter 5 is introduced for the purpose of achieving identical spectral density of the radiation, allowing for the instrument's spectral sensitivity). The parallel beam passes through controllable, three-dimensional light modulator 7 and wedge-shaped interference filter 8. The spectral characteristic of this filter, which consists of elements 7 and 8, is set equal to

$$T(\lambda) = Cf_1\left(x_1^{(1)} + \frac{\lambda - \lambda_1^{(1)}}{\lambda_1^{(2)} - \lambda_1^{(1)}}\left(x_1^{(2)} - x_1^{(1)}\right)\right), \lambda_1^{(1)} \leqslant \lambda \leqslant \lambda_1^{(2)},$$

$$\cdots \tag{5}$$

$$T(\lambda) = Cf_n\left(x_n^{(1)} + \frac{\lambda - \lambda_n^{(1)}}{\lambda_n^{(2)} - \lambda_n^{(1)}}\left(x_n^{(2)} - x_n^{(1)}\right)\right), \lambda_n^{(1)} \leqslant \lambda \leqslant \lambda_n^{(2)},$$

where $T(\lambda)$ = dependence of the energy coefficient of transmission on the optical filter's wavelength λ; C = an arbitrary positive constant. From this it is obvious that if the radiation spectrum of some "point" in the end face of bundle 4 contains n monochromatic lines corresponding to wavelengths $\lambda_1, \lambda_2, ..., \lambda_n$, representing in encoded form the point in the area of change of the unknowns with coordinates $x_1, x_2, ..., x_n$, after the passage of the radiation through the optical filter, the

brightness of the point that is optically conjugated with the one indicated above that is located on the working area of radiation receiver 10 will be

$$D = E(f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)), \tag{6}$$

where E = a constant depending on the properties of all the optical computer's elements. Thus, collimating lens 6 and focusing lens 9, along with wedge-shaped interference filter 8 and three-dimensional light modulator 7 (which are located between them), carry out the functional transformation of the set of n numbers $(x_1, x_2, \ldots, \ldots, x_n)$ into a function of n variables: $F(x_1, x_2, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n)$. Let us mention here that in connection with this, there is a change from one form of encoding to another: although before this the information about the value of a specific number $(x_1,$ for example) is completely determined by the radiation's wavelength according to (3), after passing through the interference filter the information about the value of the function and the variables will be determined completely by the brightness of the luminosity of the corresponding point, in accordance with (6).

The radiation receivers, which are made (for example) in the form of a photoconductor-electro-optical material (FPEM) structure and a readout channel (the receiver's readout channel is not shown in the diagrams in Figures 1a and 1b), make it possible to detect and register those image points for which the brightness has the given level.

The process for developing the outlines of images in which the brightness has the given level is described in [1]. Essentially, either a Bravais compensator or an oriented $\lambda/4$ plate is placed in the path of the reading beam. In connection with this, the plane of polarization of the reading beam's electrical vector through such an angle that the points with the given level of brightness will turn out to be either completely dark or the brightest points during reading. These points can be selected visually when the image being investigated is projected on a screen with sufficiently large dimensions. Let us also mention here that in this case, without any loss of time, it is possible to use receivers with sequential reading, such as a transmitting cathode-ray tube with an electronic processing unit that selects the points with the given signal level. Setting the cutoff level in the receiver at

$$D_0 = AE, \tag{7}$$

we find that on the receiver's output screen will be illuminated only those points for which the following relationship is fulfilled:

$$f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) = A, \tag{8}$$

that is, points that satisfy equation (1).

Thus, the receiver described above performs the operation of selecting those numbers for which

$$A - \varepsilon < Y < A + \varepsilon, \tag{9}$$

is correct, where ε depends on the specific parameters of the system.

In order to solve the system of equations, it is sufficient to register, in sequence on a reversible photographic film, first all those points representing the solution of the first equation in encoded form, then those for the second equation on the next frame and so on. After this all M frames (M = the number of equations

in the system) are lined up right next to each other. This stack will be transparent at those points where each of the frames is transparent. Since each frame is transparent at the points representing the solution of one of the equations in the system in encoded form, the points of transparency of the entire stack will represent (again, in encoded form) the solutions of the system of equations. Let us mention here that if in connection with this there appear distortions caused by the significant thickness of the photosensitive layers' carriers, this undesirable effect can be eliminated by sequential projection of the frames onto each other with the help of M - 1 reproducing lenses.

There is another possible way to solve such a system of equations that involves the use of FPEM structures with an electrostatic memory as a screen during the reading process. By projecting the solutions of all the equations on such a screen in sequence, it is possible to select the unknown solutions of the system (receivers based on an FPEM structure with a memory make it possible to add and subtract images). The mathematical meaning of the operations indicated above is the determination of the intersection of M sets of numbers.

In order to solve an equation of the type of (2), the method described above is first used to produce the intensity distribution described by formula (6) on an FPEM structure with an electrostatic memory, after which there is placed in front of the receiver's working surface a layer of photochromic material 14 and optical filter 15 and an additional channel consisting of light source 11, lens 12 and light-splitting filter 13. Light splitter 13 reflects the emissions of light source 11 and transmits the emissions of light source 1; photochromic layer 12 is sensitive to the emissions of source 1 and insensitive to those of source 11; optical filter 15 transmits the emissions of source 11 but not those of source 1. The spectral characteristic of the optical filter consisting of modulator 7 and interference filter 8 is set at:
in the interval $\lambda_1^{(1)} \leqslant \lambda \leqslant \lambda_1^{(2)}$ --

$$T(\lambda) = H - \frac{1}{kJ_0} \ln \varphi \left(x_1^{(1)} + \frac{\lambda - \lambda_1^{(1)}}{\lambda_1^{(2)} - \lambda_1^{(1)}} \left(x_1^{(2)} - x_1^{(1)} \right) \right), \qquad (10)$$

in the interval $\lambda_2^{(1)} \leqslant \lambda \leqslant \lambda_2^{(2)}$ --

$$T(\lambda) = P - \frac{1}{kJ_0} \ln \beta \left(x_2^{(1)} + \frac{\lambda - \lambda_2^{(1)}}{\lambda_2^{(2)} - \lambda_2^{(1)}} \left(x_2^{(2)} - x_2^{(1)} \right) \right),$$

and for all other λ, T = 0.

Here, H, P = arbitrary positive constants; J_0 = intensity of the radiation propagating in a single elementary fiber.

As is well known [6], the transmission coefficient γ of the photochromic layer is determined by the expression

$$\gamma = Le^{-kI}, \qquad (11)$$

where L, k = constants characterizing the photochromic layer; I = intensity of the incident light. Comparing formulas (11) and (10), it is easy to see that the distribution of the radiation's intensity on the receiver's working area will be

$$D = G\varphi(x_1)\beta(x_2), \qquad (12)$$

where G = a constant depending on the system's parameters.

34

From what has been said it is clear that the computer depicted by the diagram in Figure 1b makes it possible to perform the operation of multiplying several functions with different variables.

Calibrating the system so that

$$G = E \tag{13}$$

(E = the constant from formula (6)), we find that the total distribution of the transmission coefficient of the receiver's working surface is

$$D' = G(f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) + \varphi(x_1)\beta(x_2)). \tag{14}$$

Setting the cutoff level in the receiver at

$$W = BG, \tag{15}$$

we find that only those points satisfying equation (2) will be registered at the receiver's output.

Let us evaluate the basic parameters of the optical computer. The number of points processed simultaneously by the computer depends on the resolution of the optico-electronic system and the dimensions of the image field within the limits of which the optical part of the system produces no significant distortions. The optical system's resolution is limited by the receiver and the fiber-optics bundle, the resolving powers of which are approximately the same and are R ≈ 100 lines/mm [1,7]. The size of the image field can be taken as ℓ ≈ 100 mm (in connection with this it is assumed that the length of the system along the optical axis does not exceed 1.5 m). In this case the total number of points from the area of change of the unknowns among which the solutions of an equation of the type of (2) are sought will be $R^2\ell^2 = 10^8$.

The amount of time τ consumed for the actual computation process is determined by the time of registration of the image by the receiver, which depends on the power of the light source, the optical system's aperture ratio, the type of equation, the receiver's threshold sensitivity and the required accuracy in the selection of points with the given brightness level. The threshold sensitivity of receivers based on FPEM's is δ ≈ 10 μJ/cm^2 [1]. It is not difficult to derive a formula for the approximate value of the light source power Q that is needed to insure relative accuracy in the selection of points on the basis of brightness K:

$$Q = \delta l^2 \eta^{-1} K^{-1} T_{max}^{-1} \tau^{-1}, \tag{16}$$

where T_{max} = average energy coefficient of transmission of the filter corresponding to the set of radiation wavelengths for which it is maximal; η = ratio of the light flow striking the interference filter to the entire light flow emitted by the light source (in connection with this it is assumed that the receiver's cutoff level for brightness D_0 is no less than $0.1 \cdot D_{max}$, where D_{max} = maximum brightness of the points in the image). As an analysis of formula (16) shows, for η = 0.01, τ = 1 s, K = 0.01, T_{max} = 0.1 and light source power Q = $100 \cdot W$, which corresponds to actually existing light sources. The accuracy with which the found sets {x_i} satisfy equation (1) is determined not only by the accuracy of the selection of points with the given brightness level K, but also by the error in setting the required spectral characteristic of the optical filter according to (5). Evaluating the effect of this factor is an extremely complicated matter. Let us mention here that the value of variable K can be reduced not only by increasing Q or τ, but also (for example)

by photographing the image in the focal plane of lens 9 (see Figure 1a) on high-sensitivity film and then projecting the obtained positive image, illuminated by a powerful light source, on the receiver's working surface.

The limitations imposed on functions $f_i(x_i)$ are caused by the spectral characteristics of the interference filters. Existing theories make it possible to calculate interference filter designs with any given spectral characteristics [8,9]. However, it is basically impossible to realize such filters because of the presence of the presence of a small amount of absorption and scattered radiation in the dielectric layers. This undesirable effect means that it is practically impossible to manufacture filters with such a large value of $|dT(\lambda)/d\lambda|$, where $T(\lambda)$ is the filter's transmission coefficient. As an analysis of the spectral characteristics of experimental models of filters shows, the maximum values of $|dT(\lambda)/d\lambda|$ for the visible band of the spectrum are $|dT(\lambda)/d\lambda|_{max} \approx (5-10)\,nm^{-1}$. Therefore, the values of the boundaries of intervals $[\lambda_i^{(1)}, \lambda_i^{(2)}]$ should be selected during the solution of each specific problem in such a fashion that $|dT(\lambda)/d\lambda|$, as computed according to (5), do not exceed the value indicated above. Differentiating equation (5) with respect to λ, we obtain the relationships

$$\left| \frac{dT}{d\lambda} \right| = C \frac{df_i(x_i)}{dx_i} \frac{x_i^{(2)} - x_i^{(1)}}{\lambda_i^{(2)} - \lambda_i^{(1)}} \leqslant \left| \frac{dT(\lambda)}{d\lambda} \right|_{max}, \quad i = 1, 2, \ldots, n, \tag{17}$$

or

$$\lambda_i^{(2)} - \lambda_i^{(1)} \geqslant C \left| \frac{df_i(x_i)}{dx_i} \right|_{max} \left| \frac{dT}{d\lambda} \right|_{max}^{-1} (x_i^{(2)} - x_i^{(1)}), \tag{18}$$

where $\left| [df_i(x_i)]/dx_i \right|_{max}$ = the maximum value of the absolute value of $[df_i(x_i)]/dx_i$ for $x_i \in [x_i^{(1)}, x_i^{(2)}]$. Knowing the length of the computer's spectral working area and using relationship (18), it is possible to establish the limitations on the computer's basic working parameters:

$$\left[x_i^{(1)} - x_i^{(2)} \right], \quad i = 1, 2, \ldots, n; \left| \frac{df_i(x_i)}{dx_i} \right|_{max}, \quad i = 1, 2, \ldots, n.$$

Let us compare the operating speed of the proposed computer with that of a modern electronic computer. Let the computer's spectral operating band be 500-700 nm; $C = 1$; $\left| [df_i(x_i)]/dx_i \right|_{max} = \left| dT/d\lambda \right|_{max}$, $i = 1, 2, \ldots, n$; $(x_i^{(2)} - x_i^{(1)}) = 10$, $i = 1, 2, \ldots, n$. From relationship (18), for the chosen set of parameters we obtain

$$\lambda_i^{(2)} - \lambda_i^{(1)} \geqslant 10\,nm, \quad i = 1, 2, \ldots, n. \tag{19}$$

Since the extent of the computer's working zone is 200 nm and the spectral interval for the encoding of information on each unknown is at least 10 nm, the maximum value of n in this case is 20. During the solution of a problem by an electronic computer, all $f_i(x_i)$ will be represented as their expansions according to a complete, orthogonal system of functions into (for example) Fourier series. Analysis shows that for the chosen values $\left| [df_i(x_i)]/dx_i \right|_{max} = 1, 2, \ldots, n$ and C, the trigonometric series representing $f_i(x_i)$ must contain at least 10 harmonics with respect to sines and cosines. As is known, the calculation of a single value of a trigonometric function requires about 50 operations. In order to calculate the value of the left side of equation (1) for a single point, an electronic computer requires $20 \times 10 \times 2 \times 50 = 2 \cdot 10^4$ operations. For a computer operating speed of 10^6 operations per second, the time required to check a single point from the area of change of the unknowns is about $2 \cdot 10^{-2}$ s. The proposed computer requires about 10^{-8} s for this same

procedure, without allowing for the reading time, but lags considerably behind an electronic computer in accuracy and versatility. As an example illustrating this method, let us discuss the problem of processing the results of the measurement of a structure of spectral lines. The purpose of the processing is to determine the values of the wavelengths corresponding to the components of a line in the case where the spectral width of a line's structure is commensurate with the width of a spectral instrument's equipment circuit. Such a problem arises, for example, during investigations of the strengths of magnetic and electrical fields on the basis of Zeeman and Stark effects. Let the measured structure of lines contain n components of identical intensity and the spectral instrument have a (Lorents) equipment circuit:

$$D(\lambda) = 1/[1 + ((\alpha_{max} - \alpha)/\Delta)^2], \tag{20}$$

where α = wavelength of the radiation; Δ = half-width of the equipment circuit; α_{max} = wavelength corresponding to the instrument's transmission maximum; this wavelength changes during scanning. As a result of measurements, the value of the signal for n values of α_{max} is determined. During the processing it is necessary to find the values of wavelengths $\alpha_1, \alpha_2, \ldots, \alpha_n$ that correspond to the individual components of the structure. Thus, the problem reduces to solving the system of equations

$$\sum_{p=1}^{n} \frac{1}{1 + [(\alpha_{max,l} - \alpha_p)/\Delta]^2} = C_l, \quad l = 1, 2, \ldots, n, \tag{21}$$

which coincides structurally with equation (1). However, since all $f_i(\alpha_i)$ have the same form in this case, in order to encode information about $\alpha_1, \alpha_2, \ldots, \alpha_n$ it is possible to select the same spectral interval $[\lambda_1, \lambda_2]$ and a single area of change for all the unknowns: $[\alpha_{max,1}, \alpha_{max,n}]$. According to formula (5), the spectral characteristic of the interference filter for the solution of the l-th equation in system (21) should have the form

$$T(\lambda) = C/[1 + ((\lambda_{e,l} - \lambda)/\Lambda_e)^2], \tag{22}$$

$$\lambda_{e,l} = [\lambda_1 (\alpha_{max,n} - \alpha_{max,1}) - (\lambda_2 - \lambda_1)(\alpha_{max,l} - \alpha_{max,1})](\alpha_{max,n} - \alpha_{max,1})^{-1},$$

$$\Delta_e = \Delta(\lambda_2 - \lambda_1)/(\alpha_{max,n} - \alpha_{max,1}).$$

A spectral characteristic of the type of (22) is possessed by dielectric narrow-band filters made of alternating layers with high and low refractive indices in which the optical thickness of all layers except the middle one is $\lambda_{e,l}/4$ and that of the middle layer is $\lambda_{e,l}/2$ [10]. Using filters of this design, which are distinguished by the value of $\lambda_{e,l}$, it is possible to find solutions of all the equations in (21) in sequence. When solving system of equations (21) on an electronic computer for n = 10, the checking of one point from the area of change of the unknowns requires about $3 \cdot 10^3$ operations, or about $3 \cdot 10^{-3}$ s, and the checking of 10^8 points from the area of change of the unknowns requires about $3 \cdot 10^5$ s for the computations themselves. In the proposed method, the time expended directly on the computation process (without allowing for the preparatory operations) consists of the time required to read and photograph the 10 images corresponding to the 10 equations in system (21) and the time required to read the image formed by superposition of these 10 frames. It is obvious that this time will be considerably less than the time needed for the computations when the problem is solved with an electronic computer.

This article is of a purely problematical nature. The practical realization of the proposed systems will meet with considerable difficulties. However, developmental work in this direction is advisable.

BIBLIOGRAPHY

1. Balkanski, M., and Laleman, P., editors, "Fotonika" [Photonics], Moscow, Izdatel'stvo "Mir", 1978.

2. Grenot, M., Pergale, J., Donjon, J., and Marie, G., "New Electro-Optic Light Valve Device for Image Storage and Processing," APPL. PHYS. LETT., Vol 21, No 3, 1972, p 83.

3. Khass, G., and Tun, R.E., editors, "Fizika tonkikh plenok" [Physics of Thin Films], Moscow, Izdatel'stvo "Mir", Vol 5, 1972.

4. Stolov, Ye.G., "Using Wedge-Shaped Interference Filters and the Dispersing Elements of Spectral Instruments," OMP [expansion unknown], No 1, 1980.

5. Gurevich, S.B., editor, "Prostranstvennyye modulyatory sveta" [Three-Dimensional Light Modulators], Leningrad, Izdatel'stvo "Nauka", 1977.

6. Preston, K., "Kogerentnyye opticheskiye vychislite'nyye mashiny" [Coherent Optical Computers], Moscow, Izdatel'stvo "Mir", 1974.

7. Tideken, R., "Volokonnaya optika" [Fiber Optics], Moscow, Izdatel'stvo "Mir", 1975.

8. Stolov, Ye.G., "Designing Interference Optical Coatings With a Given Spectral Characteristic," OPTIKA I SPEKTROSKOPIYA, Vol 46, No 6, 1977.

9. Furman, Sh.A., and Stolov, Ye.G., "Synthesizing Multilayer Optical Coatings," OPTIKA I SPEKTROSKOPIYA, Vol 36, No 3, 1974.

10. Furman, Sh.A., "Tonkosloynyye opticheskiye pokrytiya" [Thin-Layered Optical Coatings], Leningrad, Izdatel'stvo "Mashinostroyeniye", 1977.

SOFTWARE

FUNCTIONAL FEATURES OF PROBLEM-ORIENTED COMPLEXES BASED ON SM-1410

[Article by Vitaliy Petrovich Klimenko, candidate of physicomathematical sciences; Viktor Dmitriyevich Losev, engineer; Solomon Beniaminovich Pogrebinskiy, doctor of engineering sciences; and Yuliy Samuilovich Fishman, candidate of physicomathematical sciences; all from the SKB MMS IK AN USSR [Special Design Bureau for Mathematical Machines and Systems, Institute of Cybernetics, UkSSR Academy of Sciences] (Kiev)]

[Text] At present, one of the main ways of raising the efficiency of use of computer hardware is to reduce the development time for problem-oriented complexes (POK). Rates for development and introduction of these complexes are a function of programmer labor productivity. Therefore, in addition to raising the level of a programming language and improving debugging facilities, the use of previously developed problem-oriented complexes that contain the equipment and software needed for a given field of application is becoming more and more widespread [1]. These complexes are incorporated in specific objects after relatively minor revision, the amount and time of which, however, depend strongly on the capabilities of the programming systems and hardware complex.

Considered in this article are the basic features of technology of development of problem-oriented complexes based on the SM-1410 which went into series production in 1981. These features allow implementing qualities of problem-oriented complexes which often have to be given up with the traditional computer base because the amount of effort is too large.

The SM-1410 complex is a dual processor computer system operating under control of a common operating system. The processors making up the complex have a different functional orientation and structure. Thus, the SM-2104 processor, traditional for the System of Small Computers, realizes the operating system of the complex, controls the equipment in the complex and solves problems programmed by using the programming facilities traditional in the System of Small Computers; but the SM-2410 processor is a hardware extension of the operating system that interprets the high-level ANALITIK language [2].

This computer system design allows, on the one hand, expanding the functional capabilities of the System of Small Computers through the apparatus of analytic transformations, implemented in the programming language, and advanced interaction, and on the other hand, extending system capabilities of the MIR computer series

39

through the use of peripherals and advanced software in the System of Small Computers. In other words, the advantages of these two families are combined in one system.

The dual processor complex allows:
--solving simultaneously two distinct problems with programs written in different languages;
--solving one problem, after distributing the functions between the processors: the SM-2410 processor handles the functions of transformation of models, analytic computations, precision computations and the interactive mode; the SM-2104 processor handles the functions of computations and direct interaction with the object of control;
--organizing dual control: the SM-2014 processor uses for control a set of numeric parameters, which are periodically replaced by the SM-2410 processor, which for optimization makes use of an analytic model of the object, and for feedback, the transducer readings from the SM-2104 control system. The analytic transformation facilities in the SM-2410 processor allow not only replacing the numeric parameters, but also changing the computation formulas, automatically or together with the researcher, i.e. in the interactive mode.

In the SM-1410-based problem-oriented complex software, there may be an explicit description of rather complex models of objects. These departures from the implicit prescription form, usual for small computers, for specification of models are governed by the mathematical and technical orientation of the SM-2410 processor to solving problems of enhanced complexity and the availability within the SM-1410 of two processors operating independently.

Describing and transforming mathematical models by the analytic facilities in a language close to the generally accepted language of mathematics often allows avoiding or removing the phase of shifting to a numeric model, the adequacy of which is poorly maintained, especially if it depends on higher derivatives and multiple integrals or contains nonlinear relations.

Since 60 to 90 percent of the errors in developing software associated with modeling usually occur because of the inadequacy of the models for the objects, successful debugging of the software is possible only in the interactive mode. Naturally, a model described in explicit form by systems of equations or other formula constructions is incomparably easier to review than the corresponding programs and, consequently, adjusting this model in the interactive mode is more efficient.

These properties are fully inherent to the program systems implemented on the MIR series computers which can be prototypes of problem-oriented complexes developed on the base of the SM-1410. These properties are governed mainly by the mathematical capabilities and system of engineering implementation of the input language for the SM-2410 processor, ANALITIK-79, oriented to solving problems of enhanced complexity generally encountered in scientific and engineering research.

The language is oriented to describing numeric-analytic methods in both the automatic and the interactive modes. Among its information objects (i.e. items that can be values of variables) are expressions of mathematical analysis, logic expressions and expressions consisting of character strings. The variables included in

40

these structures may also have such expressions as values. The capability of this recurrent notation allows constructing data structures that are compact and easy to computer when necessary. Language objects also include decimal numbers of unrestricted length and rational fractions. Language statements provide for a broad set of standard transformations on all information objects.

Also provided in the language is an advanced system of recognition of such functional properties as, for example, equivalency and belonging to a set, which is described in algebraic form. Although this recognition of basic mathematical forms is selective, the corresponding language procedures are defined rather generally. Thus, taken into for an expression of mathematical analysis are the properties of zero and one in operations for addition, multiplication and raising to a power. As the experience of operating the MIR series computers has shown, these facilities allow using software to construct rather efficient identifiers to automate very complex processes.

Interpretation of the programming language provides the capability of continuously controlling the information conversion process while making necessary changes to a program or data. This is especially important in solving engineering and scientific problems, when it is extremely difficult to foresee all possible alternatives. Especially efficient is the transition to the interactive mode, when the data are represented in the analytic form (for example, when files with formula elements are used in describing graphs). As experience has shown, this allows a person to easily orient himself and make the needed changes at any stage and during any cycle of operations of the problem being solved.

The high level of the language and the information capacity of its statements make the language very laconic.

Thus, according to user experiments, FORTRAN and PL-1 programs that implement numeric methods when rewritten in ANALITIK are reduced in number of statements 25 to 50 percent, and in some cases, when symbolic transformations could efficiently be used, the number of statements was reduced to one-fourth that of the original.

The relative amount of software to be developed is reduced considerably through generalization of the program concept. This is due to the fact that along with data on the problem, one can enter in the program variables and array elements, the values of which are computational formulas of the program itself (including logical). In other words, an entire class of programs with simular structure can be represented as one program with replaceable sets of formulas entering along with data.

In addition to the laconism of the input language, implementations of complex software are facilitated by the special orientation of the SM-2410 processor to fast debugging of programs. In addition to the special debugging facilities, the abundance of which is supported by hardware interpretation, the debugging process is considerably facilitated by the freedom of the input language from conventional restrictions associated with compilation facilities.

From user data and results of special experiments, it has been found that calendar periods for debugging programs on the MIR series computers are reduced three- to five-fold compared to that for systems in wide use today. When especially complex

programs are being debugged or when the work is done by specialists who are not professional programmers, this factor is even higher. This is largely due to the program being traced in the input language in the debugging process, when error diagnostics are incomparably easier that in the case of a compiled program.

It also has to be noted that the mutual orientation of the language and processor structure allowed providing rather high speed of the interpretation system.

The closeness of the input language to the conventional language of mathematics, the laconism of programs and facility of debugging allow even small groups to rather quickly design software for problem-oriented complexes with object models that are unusually complex for small computers.

BIBLIOGRAPHY

1. Stogniy, A. A. and Kurgayev, A. F., "On Problem Orientation of Computer Complexes," in "ASU: problemno-oriyentirovannyye kompleksy" [Automated Control Systems: Problem-Oriented Complexes], Kiev, UkSSR Academy of Sciences Institute of Cybernetics, 1981, pp 3-11.

2. Glushkov, V. M.; Grichenko, T. A.; Dorodnitsyna, A. A. et al., "ANALITIK-74," KIBERNETIKA, No 5, 1978, pp 114-150.

8545
CSO: 1863/37

EXPERIMENTAL SYSTEM FOR SPEECH RECOGNITION BY PHONEMES

[Article by Taras Klimovich Vintsyuk, candidate of engineering science; Ol'ga
Nikolayevna Gavrilyuk, candidate of physicomathematical sciences; Anatoliy Ivano-
vich Kulyas, engineer; Yevgeniy Kuz'mich Lyudovik, engineer; and Aleksandr
Gershevich Shinkazh, candidate of engineering science; all from IK AN USSR
[UkSSR Academy of Sciences Institute of Cybernetics] (Kiev)]

[Text] Introduction. The problem of developing efficient facilities for inter-
acting with computers that are convenient for people is relevant in connection with
the widespread use of computers and various man-machine systems for data acquisi-
tion, processing and management. New possibilities are being opened with the appli-
cation of voice input of data. As the most natural and customary for people, oral
interaction is more reliable and productive that keyboard, frees the eyes and hands
and creates conveniences for the operator. Voice input of data enhances human labor
productivity and the efficiency of using computers, automated control systems,
information retrieval systems and other man-machine systems while yielding economic
and social effects as a whole.

A main stumbling block to research, development, production and introduction of
speech recognition systems is inadequate theoretical analysis of methods and algo-
rithms for speech recognition and (or) inadequate experimental testing of these
methods. Therefore, familiarity with the few experimental systems in operation is
of major importance to speech recognition system developers.

Given in this article is a general description of the experimental system for
speech recognition by phonemes (SPRR) in operation at the Institute of Cybernetics,
UkSSR Academy of Sciences. Is is called a phoneme system since the phoneme princi-
ple of recognition has been implemented. The first phase of this system was de-
veloped in the period 1973-1974; it has been operating to the full extent since
1977. This system evolved from the experimental speech recognition system that
had been in operation in the period 1971-1974; the later system differs by being a
quasi real-time system and by implementing the phoneme principle of recognition.

System Purpose and Capabilities. The system is designed for experimental study of
speech recognition algorithms and is part of an operating complex for automated
processing of graphic information [1], where display operation is controlled by
voice.

A main purpose of this system is the use of it as a modeling stand for development of speech recognition devices and systems. This is the stand used in designing and developing the device for recognizing up to 250 oral instructions based on the Elektronika-60 microcomputer.

This system was developed on the base of the BESM-6 computer. It recognizes both discretely pronounced words or phrases and continuous speech: continuous sentences made up of words in a selected vocabulary of more than a thousand words. The capability of teachning the system allows tuning it to any vocabulary and to the voice of any speaker. The auxiliary update mode allows efficient supplementing of the vocabulary, replacing individual words in it, etc.

System Structure. The system is based on the BESM-6 computer. Connected to it is a non-standard speech signal I/O unit [2], which is a 9-bit analog-code and code-analog converter. The main conversion frequency used is 12.5 kHz, though one can select other frequencies too: 33, 25, 20, 16, 10, 8 or 6 kHz. The momentary values of a microphone signal in the form of 9-bit codes-samples enter the computer. A signal of the readniess of the next code-sample is sent to the eighth bit of a peripheral interrupt register. Upon an interrrupt signal, the computer reads out the current code-measurement by using the instruction EXT 4120B, where EXT is the instruction for exchange with peripherals, and 4120B is the address of the speech signal I/O unit in the mode of speech signal input to the computer.

A speech signal is output to a loudspeaker by using interrupt signals from the eighth bit and the instruction for output of the sample, EXT 0060B, where 0060B is the address of the speech signal I/O unit when a speech signal is output.

A communication button has a special role: a signal enters the computer only when it is pushed. Therefore, in all cases, signals are input to the computer like this: the speaker-operator depresses the button, pronounces a word, phrase or continuous sentence, and then releases the button. Thanks to this button, only the signals of interest to the person-operator are entered in the computer.

I/O programs are used to receive and output a speech signal into/from the computer. But selection of the operating mode for the speech signal I/O unit is governed by the front-end program. Thus, a speech signal can be output each time after input if this mode of operation is specified by the operator from the computer console. Input/output of a speech signal into/from the computer is effected in the mode of extracode of arbitrary physical actions.

After completion of input of a speech signal into the computer, the program for re-cognition of the input signal begins operation. The recognition result in the form of the running text of a word or series of words is shown on a light display; the indicator field of registers of the computer console is used as the display.

The speech signal I/O unit is an integral part of the phoneme-speech recognition system.

System software consists of programs for speech I/O, preprocessing of the speech signal, recognition training, recognition updating, recognition of words, recogni-tion of continuous speech and displaying recognition results.

The speech signal must be preprocessed before it can be used for training or recognition.

Speech Signal Preprocessing. The main purpose of preprocessing is to obtain a description defining the dynamics of the human vocal tract in the process of pronunciation. Preprocessing consists in computing the values of secondary features from a series of samples. Secondary features form the description of the speech signal that is directly used in the recognition process. In amount of information, the speech signal description is always considerably less than the initial description in the form of the series of samples. But the capability of using the description is governed by the speech signal redundancy. Two alternative descriptions are used in this system.

In the process of speech signal input into the computer, a running autocorrelation analysis is performed with an analysis window of 15 ms without overlap. The first $n \leq 11$ autocorrelation readings are computed by recurrence formulas so that with the reception of the next sample, the autocorrelation reading values are refined, but the samples themselves are not accumulated in computer memory. The autocorrelation readings, which are 15-ms cuts of the speech signal, form a speech element. Thus, a speech element is a vector, the components of which are n autocorrelation readings.

Let us denote by x_i the i-th observed speech element in a uniform discrete time with step $\Delta T = 15$ ms. Then the series $X_1 = (x_1, x_2, \ldots, x_i, \ldots, x_1)$ will define the running autocorrelation of the entire observed speech signal, and the value of $1 \cdot \Delta T$ will be the length of the analyzed speech signal. It is evident that $11 \times 67 \simeq 740$ storage cells will be needed to store the description of a speech signal with the length of 1 s.

Compression of the information in the process of analysis and computation of each speech element is shown by the fact that 187 initial samples of the speech signal are replaced by 11 readings of the autocorrelation vector (11 storage cells per speech element).

The first description alternative is characterized exactly by the fact that realizations of words and continuous speech are specified by series of elements: autocorrelation vectors.

The second description alternative is more economical: only one storage cell is needed to store one speech element x_i. In this case, element x_i is a 48-bit binary code having the meaning of the symbol of the derivative of the spectrum with respect to frequency in a discrete grid of 48 frequencies [3]. The elements-codes are computed by conversions of autocorrelation vectors.

For this, the autocorrelation vector is first converted into a so-called vector of of forecasting parameters (a-parameters) of the speech signal and white noise energy [4]. These parameters specify a linear autoregression filter that describes the vocal tract. The filter parameters and white noise energy are chosen in such a way that the white noise energy is minimal while the signals, generated by the linear filter when excited by the white noise energy, are described by the same autocorrelation vector as that for the speech element observed.

The conversion from an autocorrelation vector to the a-parameters and white noise energy is effected by using the Durbin algorithm [5] which rapidly solves a specific system of (n-1) linear equations.

Let $a = (a_1, a_2, ..., a_{n-1})$ and A be the vector of forecasting parameters and the white noise energy, respectively.

Then we compute the so-called autoregression spectrum of the speech signal G(p) in a grid of 49 spectral frequencies

$$G(p_r) = A \left/ \sum_{s=0}^{n-1} b_s \cos sp_r\pi, \right.$$

where p_r is the relative frequency, $0 < p_r < 1$, $r = 1, 2, ..., 49$; and $b = (b_0, b_1, ..., b_s, ..., b_{n-1})$ is the so-called vector of b-forecasting parameters (b-parameters)

$$b_0 = \sum_{v=0}^{n-1} a_v^2,$$

$$b_s = 2 \sum_{v=0}^{n-s-1} a_v a_{v+s}, \quad s = 1, 2, ..., n-1;$$

$$a_0 = 1.$$

Finally, based on the autoregression spectrum, we compute the 48-bit binary element-code

$$x_i = (x_{i1}, x_{i2}, ..., x_{ir}, ..., x_{in}),$$

$$x_{ir} = \begin{cases} 1, & \text{if } G(p_{r+1}) \geq G(p_r) \text{ and } G(p_{r+1}) \geq \theta_r, \\ 0 & \text{in other cases,} \end{cases}$$

where θ_r, $r = 1, 2, ..., 48$ are some constant thresholds chosen experimentally in such a way that under stationary conditions, zero elements-codes are generated when only acoustic noises of the room enter the microphone input.

Series of elements-codes substantially more economically describe a speech signal than series of elements-vectors of autocorrelations. Nevertheless, the elements-codes contain sufficient information necessary for recognition. Thus, they specify the dynamics of the motion of spectral maximums and minimums, contain information on the qualities of the poles of the vocal tract, etc.

Use of the technique of computing the codes through an autoregression spectrum [6] stems from the desire of achieving the aims through minimal computations to save time. Thanks to this technique, it has become possible to first accumulate the values of the autocorrelation vectors directly in the process of speech signal input, and then rapidly convert autocorrelation vectors to elements-codes, which can be done after the end of the input. This conversion requires only 0.25 s per 1 s of input speech signal.

Thus, preprocessing for the descriptions used is performed basically in the speech signal input process, i.e. in real time.

But using the traditional technique of the Fast Fourier Transform inevitably leads to the necessity of storing speech signal samples in computer storage, to discontinuing the use of computer resources when the signal is input, and to a three- to five-fold delay as a whole in speech signal preprocessing.

Further, the accumulated realizations $X_l = (x_1, x_2, \ldots, x_i, \ldots x_l)$ of the speech signal in the form of series of autocorrelation elements or elements-codes go for recognition or recognition training. Training always precedes recognition. But it is convenient to begin describing the training and recognition procedures with recognition.

Recognition of Words of Speech. Assume recognition training has already been completed. Then there will be two sets of values in computer storage.

The first set, E, consists of 80 standard elements, $e(j) \in E$, where j is the name, the ordinal number of the element $e(j)$ in set E. Just as the observed elements x_i, the standard elements $e(j)$ are (describe) 15-ms cuts of the speech signal. But in contrast to x_i, the elements $e(j)$ are type elements of speech. They are chosen so that they best approximate the speech elements being observed. In content, the standard elements are phonemes and their parts also characterize individual spunds of speech by method and place of formation.

In the case of using elements-codes, standard elements $e(j)$ are also elements-codes. In this case, 80 storage cells are needed to store E.

In the case of using elements-vectors of autocorrelations, the standard elements are b-parameters, and then 80 X 11 cells are needed to store set E.

The second set of pairs of transcriptions $(R_k \tau_k)$, k=1, 2, $\ldots$, K, specifies the acoustic R_k and temporal τ_k transcriptions for all the words in the vocabulary, where k is the number of the word in the vocabulary and K is the size of the vocabulary.

The acoustic transcription of a word, $R_k = (j_{k1}, j_{k2}, \ldots, j_{ks}, \ldots, j_{kq_k})$ is a series of names of elements from the standard set E. It indicates which, and in what sequence, elements must be taken from E so that in forming with respect to the transcription R_k the initial standard of the k-th word

$$R_k E = (e(j_{k1}), e(j_{k2}), \ldots, e(j_{ks}), \ldots, e(j_{kq_k}))$$

and converting this initial standard be nonlinear deformation along the time axis, a good approximation of the different realizations of the k-th word can be achieved by the converted standards.

Limitations on the possible deformations of the initial standard of the word are specified by its temporal transcription

47

$$\tau_k = ((m_{k1}, M_{k1}), (m_{k2}, M_{k2}), \ldots, (m_{ks}, M_{ks}), \ldots, (m_{kq_k}, M_{kq_k}))$$

The pair (m_{ks}, M_{ks}) indicates that with nonlinear expansion of the initial standard R_kE, the element $e(j_{ks})$ may be repeated from m_{ks} to M_{ks} times. In such a nonlinear expansion (repetition of elements), the sequence of standard elements specified by the acoustic transcription must be maintained.

The temporal transcription, τ_k, specifies the set $\tau_k(1)$ of statements v $\tau_k(1)$ of the nonlinear expansion of the initial standard R_kE with length q_k to length 1. The action of these statements is described thus:

$$vR_kE = (\underbrace{e(j_{k1}), \ldots, e(j_{k1})}_{v_1 \text{ times}}, \underbrace{e(i_{k2}), \ldots, e(j_{k2})}_{v_2 \text{ times}}, \ldots,$$

$$\underbrace{e(j_{ks}), \ldots, e(j_{ks})}_{v_s \text{ times}}, \ldots, \underbrace{e(j_{kq_k}), \ldots, e(j_{kq_k})}_{v_{q_k} \text{ times}}),$$

where

$$v = (v_1, v_2, \ldots, v_s, \ldots, v_{q_k}), \quad m_{ks} \leqslant v_s \leqslant M_{ks}.$$

$$s = 1, 2, \ldots, q_k; \quad \sum_{s=1}^{q_k} v_s = l.$$

Thus, we have defined a certain process for generating standard signals of words of arbitrary length from the set of standard elements common for all words in accordance with the acoustic and temporal transcriptions individual for each word.

It is evident that the various standard signals of a word vR_kE, $v \in \tau_k(1)$ differ nonlinearly by the changing rate of pronunciation. This is how the basic factor of variability of speech signals is taken into account. The standard signals vR_kE are coarticulated and reduced signals. Thus, in the transcription R_k, there are the so-called transition elements for which $m_{ks} = M_{ks} = 1$, i.e. these elements cannot be repeated with deformations. In transcriptions R_k, the first and last elements are always pause elements. For them

$$m_{k1} = m_{kq_k} = 0; \quad M_{k1} = M_{kq_k} = \infty.$$

This means the length of the pause at the beginning and end of words may be any, including zero. For internal elements, m_{ks} is always ≥ 1.

As a whole, this process of generating standard signals of words takes into account the basic factors of variability of speech. Therefore, this process should afford good approximation of realizations.

Transcriptions may contain from 4 to 20 elements; thus, 6 storage cells are needed to specify pairs of transcriptions (R_k, τ_k) of a word.

During recognition of a presented realization $X_1 = (x_1, x_2, \ldots, x_i, \ldots, x_1)$ with length 1, standard signals of words of the same length are generated and compared to the realization presented. The response $k(X_1)$ of the recognition of the

realization X_1 declares the word, the standard signal of which is found most similar to that being recognized:

$$k(X_l) = \operatorname*{argmax}_k \max_{v \in \tau_k(l)} G(X_l, vR_kE) =$$

$$= \operatorname*{argmax}_k \max_{v \in \tau_k(l)} \sum_{i=1}^{l} g(x_i, (vR_kE)_i).\qquad(1)$$

where $G(X_1, vR_kE)$ is the similarity of the X_1 being recognized and the standard signals, and $g(x_i, (vR_kE)_i)$ is the elementary similarity of the x_i, being recognized, and the standard $(vR_kE)_i$ elements. The latter is in the i-th position in the series vR_kE.

The following values:

$$g(x_i, e(j)) = -H(x_i, e(j))$$

are selected as the elementary similarity $g(x_i, e(j))$, where $H(x_i, e(j))$ is the Hamming distance between the codes x_i and $e(j)$ when elements-codes are used as the speech signal description [7]; and

$$g(x_i, e(j)) = -(x_i, e(j))/x_{i1}^{3/4},$$

where $(x_i, e(j))$ is the scalar product, and X_{i1} is the first component of the vector of the autocorrelation x_i (energy of the speech element) when autocorrelation vectors are used as the speech elements [8].

Of course, in the process of recognition by criterion (1), generation of all possible standard signals vR_kE and comparison of each of them with the realization to be recognized do not occur: no computer resources are sufficient. Comparison of the realization X_i with standards vR_kE of word k, finding the standard vR_kE with the greatest similarity and calculating the value of this similarity are performed by directed search by using the special scheme of dynamic programming [9, 10] which essentially makes use of the specifics of the problem being solved (1).

Thus, computation of

$$G_k(X_l) = \max_{v \in \tau_k(l)} G(X_l, vR_kE)$$

is realized by using the method of dynamic programming, and the search for the optimal

$$k(X_l) = \operatorname*{argmax}_k G_k(X_l)$$

is performed by exhaustive search.

Thus, to recognize a realization of a word from a vocabulary of K words, K problems of dynamic programming have to be solved each time. The recognition result is shown in text form on a light display.

Recognition response delay after the end of word pronunciation for a 100-word vocabulary is 0.6 s; for a 500-word, 4 s; and for a 1000-word, 8 s.

Reliability in recognizing words pronounced separately for vocabularies of 100, 500 and 1,000 words is more than 99.5, 98 and 96 percent, respectively. In this case, recognition is selecting one hypothesis from 100, 500 or 1000.

These data on recognition reliability and response delay are identical for both methods of describing the speech signal.

Maximum duration of signals of words to be recognized taking into account pauses at the beginning and end of the word is limited to 2.25 s.

Manifestation of the Phoneme Principle. Without analyzing the degree of manifestation of the phoneme principle in the system described, let us note only that its implementation yields considerable savings in storage and volume of computations.

Thus, in the device made by the Nippon Electric Co. [11], the source standards of words contain on the average not 12, as in our case, but 50 elements. With that, the standard elements of a word are individual for each word. Storing standards of words from a vocabulary of K words requires 50nK storage cells compared to the 80n+6K cells for our case (80n cells for storing the set of standard elements and 6 for storing the transcriptions). We have about an 80-fold advantage in storage for n = 11 and K = 1000.

The advantage in computation volume is achieved primarily because for each element x_i to be recognized, we have to consider not 50K elementary similarities $g(x_i, e_i)$, but just 80, i.e. just the standard elements in set E. Thus, when K = 1000, we have more than a 600-fold advantage in computing similarities.

Further reduction in computation is achieved because the problems of dynamic programming are solved with an average grid size of 12X1, and not 50X1, as with the Nippon device.

Speech Recognition Training. This process begins with accumulating a training sample of u, u = 1 - 10, realizations for each word. Gathering of a sample is attended by indication of the text of the word and length of the transcription q_k of each word, which is usually 1.5-fold longer than the length of the phonetic transcription of the word. The length q_k is specified either manually from the computer console or automatically computed from word text.

Evaluated together in the training process are the set E, common for all words, of N, N $\leq$ 80, standard elements, and the acoustic R_k and temporal τ_k transcriptions of each word. These parameters are selected so that the standard signals drawn from E for the transcriptions R_k and τ_k best approximate the realizations of the training sample, i.e. the criteria

$$\sum_{u=1}^{L} \max_{v_u \in \tau_{k(u)}(l^u)} G(X_u^u, v_u R_{k(u)} E),$$

are maximized, where k(u) is a function indicating which class the realization with the number u belongs to, and L is the quantity of realizations in the training sample.

The training task (it may also be called the self-training task) is integral to a considerable extent. The algorithm for handling it, described in [7] in detail, stipulates that one of the standard elements in E must be a pause element, and all transcriptions R_k begin and end with a pause element with limitations on repeatability $(0, \infty)$.

It takes no more than two hours to train the system for a vocabulary of 200 words with five realizations per word taking into account accumulation of the sample on magnetic tape.

Speech Recognition Finish Training. This procedure is intended for adding to the vocabulary and replacing individual words in the vocabulary. The already known set E of standard elements is considered in finish training. A training sample of 1 to 10 realizations of a word is accumulated, and from it only the acoustic R_k and temporal τ_k transcriptions of the new word are evaluated [7]. Finish training with five realizations per word takes about 15 s.

The following mode of operation is typical: First, the system is trained to recognize 200 words, and then another 800 words in finish training. The results of training and finish training are used equally for recognition of continuous speech as well.

Recognition of Continuous Speech. It is assumed that the continuous speech is from the vocabulary taught to the system. Word order in continuous speech is considered free. It is evident that under these conditions, there will be 1000^5 possible phrases of five words (when there are 1000 words in the vocabulary).

In recognizing continuous speech, the signal presented is compared to standard signals of continuous speech that are derived as a result of combining into a series of standard signals the individual words composed from E with regard to the transcriptions R_k and τ_k. A special dynamic programming routine divides the continuous speech signal into separate parts corresponding to words, performs the recognition of these parts as words and organizes the directed search for the optimal solution: the series and number of words with indication of optimal bounds between words that is achieved by the greatest similarity between the series of standard word signals and the signal being recognized. All features of the continuous speech recognition algorithm are described in [12, 9]. The continuous speech recognition result is the series of words contained in the signal being recognized.

The maximum duration of phrases to be identified is eight seconds. This means that there may be up to 15 words in the continuous phrases.

Continuous speech recognition reliability in terms of word recognition is 93 percent with 200 words in the vocabulary (branching factor of 200).

The tediousness of computations in continuous speech recognition in terms of one second of speech is about 1.1-fold higher than in word recognition. Recognition response delay for a five-second cut of continuous speech with 200 words in the vocabulary is, thus, about 12 s.

Fig. Software Structure

Key:
1. parameters
2. accumulation of training sample
3. segmentation of training sample
4. taxonomy
5. training
6. finish training
7. recognition of isolated words
8. recognition of continuous speech
9. display
10. supervisor
11. computer console

Software Structure. The system software
contains eight independent modules that
alternate in computer storage (figure).
The operator calls in the module or
block of modules needed through the
console. The supervisor control
program provides for interaction
between modules.
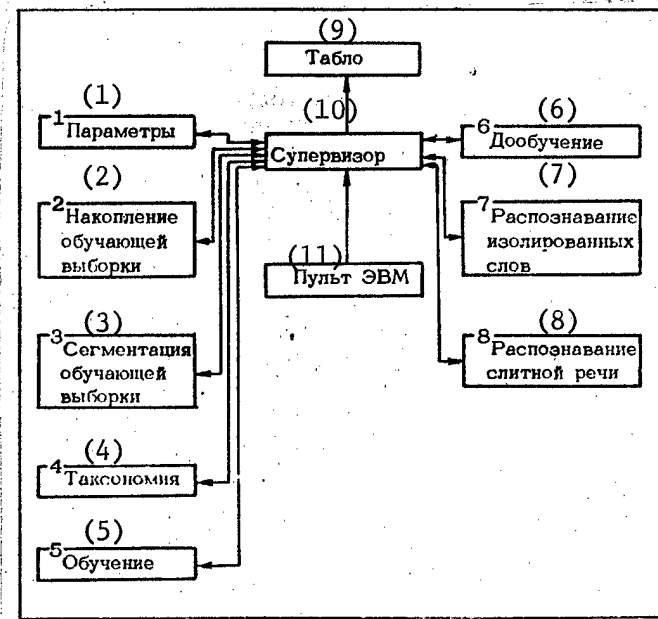
A component of modules 1, 2, 6, 7 and 8 is the speech signal preprocessing program,
parameters of which are the quantity n of autocorrelation readings, length of
analysis interval, frequency of sampling and others. These parameters are speci-
fied from the console. Module 1 is used to select the thresholds needed to
compute elements-codes.

The first five modules are used in training. Modules 3 and 4 are auxiliary with
respect to module 5. They find the initial approximation for the iteration train-
ing algorithm. The function of the other modules is evident from their title.

All user interaction with the speech recognition system is performed through the
computer console; during operation, it is used to select a particular mode,
specify the number of words to be recognized, the number of a word to be corrected,
etc.

The software is compiled primarily in the MADLEN language. Control programs are
written in FORTRAN.

The experimental system for speech phoneme recognition is an evolution of the
efforts on speech recognition performed in the period 1967-1973, which were based
on making up standard signals from elementary parts and on the application of
dynamic programming [10, 13-15]. However, implementation of the phoneme principle
and procedures for teaching phoneme-by-phoneme recognition, and use of special
computational dynamic programming schemes that consider time restrictions have
allowed developing and implementing in the current quasi-real-time system new
highly efficient speech recognition methods that require substantially less
storage and computations than the other similar methods based on dynamic program-
ming that have emerged in the USSR and abroad within the last five years.

Modules for generalized recognition and understanding of continuous speech
are now being added to the experimental system for phoneme-by-phoneme recognition
[16].

BIBLIOGRAPHY

1. Vintsyuk, T. K. and Shinkazh, A. G., "Automated Graphic Information Processing
    System Using a Speech Recognition System and Display," in "Avtomaticheskoye
    raspoznavaniye slukhovykh obrazov" [Automatic Acoustic Pattern Recognition],
    Tbilisi, Metsniyereba, 1978, pp 178-180.

2. Vintsyuk, T. K., "Efforts by the UkSSR Academy of Sciences Institute of Cyber-
    netics on Automatic Speech Signal Recognition," in "Raspoznavaniye obrazov.
    Trudy mezhdunar. simp. 1971 g. po prakticheskim primeneniyam metodov raspozna-
    vaniya obrazov" [Pattern Recognition. Transactions of the International Sym-
    posium on Practical Applications of Pattern Recognition Methods held in 1971],
    Moscow, VTs AN SSSR [USSR Academy of Sciences Computer Center], 1973,
    pp 106-117.

3. Vintsyuk, T. K. and Shinkazh, A. G., "Phoneme Recognition of Words of Oral
    Speech: Algorithms for Training and Recognition and Experimental Results,"
    in "Avtomaticheskoye paspoznavaniye slukhovykh obrazov" [Automatic Acoustic
    Pattern Recognition], Lvov, LGU [Lvov State University], 1974, part 3,
    pp 19-24.

4. Markel, J. D. and Gray, A. H., "Linear Prediction of Speech," Moscow, Svyaz',
    1980, 308 pages.

5. Anderson, T., "Statistical Analysis of Time Series," Moscow, Mir, 1976, 756
    pages.

6. Vintsyuk, T. K. and Kulyas, A. I., "General-Purpose, Real-Time Speech Analysis
    Program," in "Avtomaticheskoye raspoznavaniye slukhovykh obrasov," Tbilisi,
    Metsniyereba, 1978, pp 182-184.

7. Vintsyuk, T. K. and Shinkazh, A. G., "Automatic Transcribing of Patterns from
    a Teaching Sample," in "Obrabotka i raspoznavaniye signalov" [Signal Processing
    and Recognition], Kiev, IK AN USSR [UkSSR Academy of Sciences Institute of
    Cybernetics], 1975, pp 102-120.

8. Vintsyuk, T. K.; Gavrilyuk, O. N.; and Lyudovik, Ye. K., "Phoneme Recognition
    of Words and Continuous Speech by Prediction Parameters," in "Matematicheskiye
    i tekhnicheskiye srestva robototekhnikhi i raspoznavaniya obrasov"
    [Software and Hardware for Robot Technology and Pattern Recognition], Kiev,
    IK AN USSR, 1981, pp 71-79.

9. Vintsyuk, T. K.; Gavrilyuk, O. N.; and Shinkazh, A. G., "Phoneme Recognition of
    Continuous Speech Composed of Words from a Selected Vocabulary," in
    "Raspoznavaniye obrasov" [Pattern Recognition], Kiev, IK AN USSR, 1977,
    pp 34-43.

10. Vintsyuk, T. K., "Raspoznavaniye slov ustnoy rechi metodami dinamicheskogo programmirovaniya" [Spoken Word Recognition by Dynamic Programming Methods], KIBERNETIKA, No 1, 1968, pp 81-88.

11. "NEC [Nippon Electric Company] Continuous Speech Recognition System," ZARUBEZHNAYA RADIOELEKTRONIKA, No 4, 1980, pp 108-120.

12. Vintsyuk, T. K., "Element-by-Element Recognition of Continuous Speech Composed of Words from a Specific Vocabulary," KIBERNETIKA, No 2, 1971, pp 133-143.

13. Vintsyuk, T. K., "Teaching Element-by-Element Recognition of Speech," in "Raspoznavaniye obrasov i konstruirovaniye chitayushchikh avtomatov" [Pattern Recognition and Design of Reading Automata], Kiev, IK AN USSR, 1969, No 2, pp 23-35.

14. Vintsyuk, T. K., "Methods of Learning, Self-Learning and Speech Recognition Based on Compilation of Standard Signals from Elementary Parts," in "Trudy 8-y Vsesoyuznoy akusticheskoy konferentsii. Plenarnyye doklady" [Transactions of Eighth All-Union Conference on Acoustics. Plenary Papers], Moscow, AKIN AN SSSR [USSR Academy of Sciences Institute of Acoustics], 1973, pp 75-90.

15. Vintsyuk, T. K.; Gavrilyuk, O. N.; and Puchkova, N. G., "Experimental Voice Input System for Digital Computers," in "Trudy 8-y Vsesoyuznoy akusticheskoy konferentsii. Sektsiya 'Raspoznavaniye obrasov'"[Transactions of Eighth All-Union Conference on Acoustics. Section on "Pattern Recognition"], Moscow, AKIN AN SSSR, 1973, pp 65-68.

16. Vintsyuk, T. K., "Two Basic Ways of Developing Speech Recognition and Understanding Systems for Continuous Speech," in "Avtomaticheskoye raspoznavaniye slukhovykh obrasov" [Automatic Acoustic Pattern Recognition], Yerevan, 1980, pp 221-225.

PROCESS INTERACTION MECHANISM IN COMMUNICATION PROCESSOR CONTROL PROGRAM

[Article by Vladislav Vladimirovich Gusev, candidate of physicomathematical
sciences, and Anatoliy Petrovich Chernat, engineer, both from the UkSSR Academy of
Sciences Institute of Cybernetics (Kiev)]

[Text] In computer networks with packet switching, general-purpose or specialized
minicomputers are used as communication processors (KP). In both cases, the com-
munication processor control program must provide for parallel execution of packet
reception, processing and transmission to communication channels; thus, it must
include facilities for interaction of parallel processes. Operation of these
facilities requires drawing on a certain portion of the communication processor
resources (central processor [CPU] time and main storage) which are scarce in the
majority of cases.

The requirements that have to be met in selecting interaction mechanisms are con-
tradictory in the general case:
--functional completeness (they must provide for all needed types of interaction:
data exchange, access to common resources, etc.);
--convenience of use (clarity of program designs, simplicity in description of
interaction);
--efficiency (minimal diversion of communication processor basic resources to
realize the interaction mechanism); and
--flexibility (optional capability of changing and adding types of interaction
should be provided).

To meet the requirement of high efficiency in using communication processor equip-
ment, it may prove worthwhile in designing the processor control program to not
use the standard minicomputer OS with the inevitable overhead due to its generality
and instead develop a highly specialized process control nucleus that includes
the specific functions and types of communication processor process interaction
that are known in advance.

Experience in developing a communication processor control program has shown [1]
that the following types of process interaction can be included in it: trans-
mission of messages (in packet form), joint use of common resources, and synchron-
ization of processes by events.

These types of interaction are not considered fundamentally different here, and under certain conditions, unified program primitives, such as semaphores, can be drawn on to implement them. But more precise consideration of their substantive differences is needed to efficiently and adequately design the program.

Brief Survey of Process Interaction Facilities. A comparison of the known forms for describing the interaction of processes (events, semaphores, monitors, mailboxes) shows that each has certain advantages in implementing certain types of interprocess communications and is less efficient with others. For example, the apparatus of events in Unified System OS [2] provides for only the simplest form of synchronization through which a process may await execution of an event indicated by another process. A process may await completion of one of several events, but a given event may be awaited by no more than one process (a queue of waiting processes is not provided for). Because of such restrictions, other facilities (the ENQ and DEQ macroinstructions) are used for work with common resources in Unified System OS.

The Dijkstra semaphores [3-7] are convenient in working with common resources, but programming a message transfer (possible per se) produces a rather awkward construction.

Let us recall the possible interpretation of semaphores. A semaphore may be implemented in the form of a counter and queue for placing processes being delayed. The processes obtain access to a semaphore by two operations, indivisible in time, usually denoted as P and V. A process which performs a P operation on a semaphore decrements the counter by one. In this process, if the counter value is negative, the process is switched to the wait state and placed in the semaphore queue. A V operation increments the counter by one. When the new value of the counter is negative or zero, a waiting process is taken from the queue and put in the active state. The continuity of P and V operations is supported by hardware, usually by using interrupt blocking.

It is evident that the waiting by a process for an event can be modeled by using a semaphore with a simplified construction of the queue (for placing one process in it). At the same time, waiting for one of several events can not be represented by any combination of P operations on semaphores.

Mailbox facilities are often used to transfer messages between processes; a mailbox is considered a set of several slots for placing messages [5]. The sending process and receiving process access the mailbox independently of each other. When all the slots in a mailbox are filled, the message transfer operation is not executed and the sending process is notified of this by an appropriate return code. Similarly, when a mailbox is empty, a process that had issued a request to receive messages also receives a negative return code. Thus, the mailbox in this form does not provide the capability of synchronizing processes.

The Hoare monitor [8] is a rather universal and, at the same time, enlarged facility for process communication. The monitor is a set of data structures and procedures executable on them. Also, at any one time, only one process is afforded entry to the monitor (in the form of execution of a procedure). If a monitor is already busy, then the new process that called the monitor will be placed in its queue and kept until release of it.

For tasks similar to waiting for an event, there are built-in procedures for delay (delay) and continuation (continue) of processes, executable on variables of a special type "queue." Execution of a delay procedure by a process causes placement of it in the indicated queue and release of the monitor. Subsequently, this process may again be activated and become the owner of the monitor if another process executes a continue procedure applicable to the queue at the head of which it is (in this case, execution of the continue procedure is considered an abnormal exit from the monitor).

The main type of interaction between processes in the communications processor is the transfer of messages through queues of unlimited length serviced in accordance with the simplest FIFO procedure. Let us consider an implementation of such a queue in the form of a monitor in the CONCURRENT PASCAL language [9]:

```
begin
tupe UNBOUNDBUF: monitor;
var  SIZE: integer;
     RECEIVER: queue;
  .
  .
  .

procedure entry PUT(X:data);
begin
/*поместить элемент данных в очередь сообщений*/
     [place data element
                 in message queue]
  .
  SIZE:=SIZE+1;
   continue (RECEIVER);
end;
procedure entry GET (var X:data);
begin
   if SIZE=0 then delay (RECEIVER);
/*извлечь элемент данных из очереди сообщений*/
     [take data element out of
                 message queue]
  .
   SIZE:=SIZE—1;
 end;
   SIZE:=0;
  .
  .
  .
end;
```

Implementation of the same type of interaction by using semaphores also requires two process queues—for those waiting for release of a critical sector (implicit in the case of a monitor) and for those waiting for a released resource (associated in our case with the emergence of a message to be transferred)—and a data queue.

```
/*transfer of message*/
   P (CRITICAL SECTOR)

   .
   .
   .

/*place message in message queue*/
   SIZE:=SIZE+1
   V (CRITICAL SECTOR)
   V (DATA)
```

```
/*reception of message*/
    P (DATA)
    P (CRITICAL SECTOR)
    .
    .
    .
/*take message from message queue*/
    SIZE:=SIZE-1
    V (CRITICAL SECTOR)
```

It is evident that the facilities for implementing the queue of processes awaiting a resource and the data queue can be combined since at any time, only one of these queues is not empty. Also, the semaphore CRITICAL SECTOR is eliminated since hardware facilities are used that afford continuity of operations on the combined queue. The extended mechanism obtained for process interaction is known as a semaphore with data [10]. Just as before, a negative value in the counter for the semaphore with data is equal to the number of processes delayed in waiting for a resource, while a positive value is equal to the number of data elements in the queue. Operations for receiving data are designated DP, and those for sending data are DV.

Semaphores with data allow describing and implementing the task of interest to us in the simplest way:

```
/*sending of a message*/
    DV (semaphore of messages, address of message)
    .
    .
    .
/*receiving of message*/
    DP (semaphore of messages, address of message)
```

Semaphores with data allow coordinating the work of processes by exchanging messages and to a considerable extent correspond to the style of the data flow in parallel programming [10].

More practical is the task of message transfer through a queue of finite length, i.e. when there is a finite number of elements in the buffer pool for placing messages to be transferred. It is evident that in this case, the empty buffers must also be considered a resource to be sent through an additional semaphore with data from the receiving process to the source process:

```
/*sending of a message*/
    DP (semaphore of buffers, address of buffer)
    /*filling of buffer*/
    .
    .
    .
    DV (semaphore of messages, address of message)
    .
    .
    .
/*receiving of message*/
    DP (semaphore of messages, address of message)
    /*processing of message*/
    .
    .
    .
    DV (semaphore of buffers, address of buffer)
```

The degree of asynchroneity of process interaction can be indicated. Thus, when there is but one buffer in the pool (initial value of the counter for the semaphore of buffers + 1), it is minimal since a new transfer cannot be organized until the receiving process has received the preceding message.

The evident symmetry in describing the actions of the source process and the receiving process in the example cited above can be the basis for introducing an enlarged facility for describing process interaction which is a combination of two semaphores with data [11]. But as applied to a communications processor control program, this solution seemed unwarranted. In this case, the common buffer pool would have to be divided into non-intersecting sub-pools in accordance with the number of semaphores with data used for transferring packets from process to process through a rather complex scheme and additional operations for an overload of packets would have to be executed.

Combined Waiting. Processes in a communications processor control program that execute the specific function of processing packets receive them through an input queue (in the form of a semaphore with data). In the course of processing, they may need additional resources (say, free buffers). Requests for packets and buffers are usually issued sequentially, but in some fundamentally important cases, there is a need to realize the state of waiting by a process for any of two (or more) types of objects. A possible solution may be organizing the program so that certain generalized objects are transferred to the receiving process through a common input queue. These generalized objects may be either the objects proper that contain an additional indicator of type (message or buffer) in the heading or references to a non-empty queue (message or buffer queue) [12]:

```
                    /*receiving of generalized object*/
        DP (semaphore of references, ref)
        case ref of
          message: begin
            DP (semaphore of messages, address of message);
            /*processing of message*/
              .
              .
              .
                end;
          buffer: begin
            DP (semaphore of buffers, address of buffer);
            /*filling of buffer*/
              .
              .
              .
                end;
        end case;
```

But in this solution, in addition to the introduction of the inappropriate task of an additional stream of references, there is another shortcoming that is related to the assumption of uniformity of lists of requests for resources for all processes. This circumstance excludes the possibility of using it for the task mentioned above with a general pool of buffers when, for example, free buffers are being requested from the pool by some processes, while others (along with similar requests) are concurrently awaiting the arrival of a packet (and in the case of receiving a packet, the buffer request has to be abandoned).

The problem of concurrent waiting also arises when there is a need to describe the waiting, restricted in time, by a process for allocation of some resource.

More adequate problems being described should consider introduction of the extended operation P which allows waiting with several semaphores, combined if need be with waiting by time.

Generalized Semaphore. Let us call a generalized semaphore the data structure needed to organize combined waiting by a process for one of several events or types of resources. For the generalized semaphore, let us introduce the operations GP for a request and GV for a transfer.

A GV operation is similar to the V and DV operations discussed earlier:

<send operation>::=GV<variable-semaphore>
    [from <variable-indicator>]

The GP operation is generalized for the case of waiting with several semaphores, combined, perhaps, with waiting by time. Implementation of a particular awaited event leads to selecting a program branch to be executed, which it is also advisable to include in the syntax for the GP operation:

<request operation>::=GP<alternative>{;<alternative>} [delay<delay>] end

<alternative>::=<variable-semaphore>[into <variable-indicator>]: <compound
    statement>

<delay>::=<arithmetic expression>:<compound statement>

The information sent through the semaphore queue is intentionally limited to an indicator type expression to ensure efficiency in implementation. It is also assumed that the variable-indicator contains a reference to the data structure having the necessary fields for the adopted method of including it in the semaphore queue.

Implementation of Generalized Semaphores. In developing the communication processor control program, an approach was taken that allowed, within the bounds of a unified process description method, ensuring implementation of the most suitable type of interprocess communication in each specific case.

Queues of objects (processes or messages), associated with a generalized semaphore, may have a different structure, including one in which no more than one object may be in a queue. To this end, in the block for controlling a generalized semaphore, there may be indicated different procedures for setting and taking objects from a queue, which allows modeling semaphores with data in some cases and events in others.

The capability of a process being in queues of several generalized semaphores at the same time to implement the status of combined waiting is provided by the presence of several communication blocks in the process control block. Their number is chosen for each process statically at the stage of program development, based on the maximum number of operands in the GP operations available. The number of of communication blocks used is fixed in the waiting counter in the process control block.

The communication block contains fields for reference to the next and preceding
communication block in the chain, as well as the address of the process control
block associated with it and the branch address, which corresponds to the address
of the branch in the program to be executed in case of implementation of the
appropriate event.

Shown in the figure is a simplified
diagram of implementation of
communications between semaphores and
processes for the case when one process
performs waiting with semaphore 1,
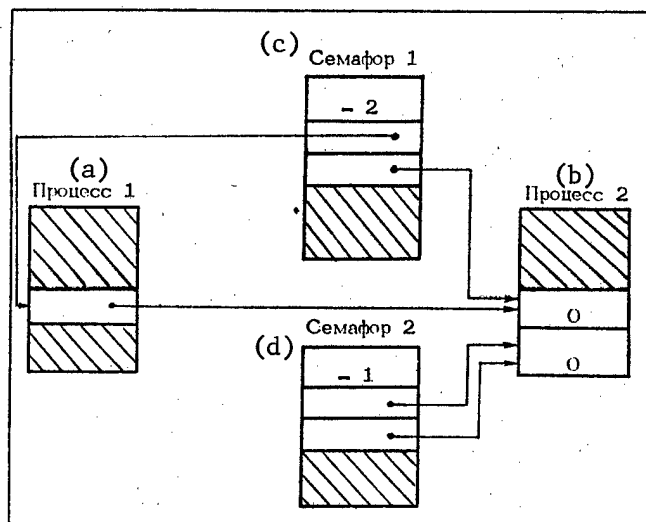while another, with semaphores 1 and 2
at the same time.

Here are the algorithms for the GP and
GV operations:


```
            /*GP operation*/
begin
  I:integer:=1;
  close interruption;
  while list of semaphores is not
                 exhausted loop
    semaphore counter minus 1;
    if    semaphore counter<0
    then  include communication block I
          in semaphore queue I;
          I:I+1;
    else  delete communication blocks from 1 to I-1 from semaphore queues from 1 to
          I-1;
          increase semaphore counters from 1 to I-1;
          delete message from semaphore queue I and send message to process;
          set start address to branch address;
          go to EXIT;
    end if;
  end loop;
  waiting counter:=I;
  take next active process for execution;
  EXIT:  open interruption;
end;
```

Key:

a.	process 1	c.	semaphore 1
b.	process 2	d.	semaphore 2

```
                      /*GV operation*/
begin
  close interruption;
  semaphore counter plus 1;
  if    semaphore counter>0
  then  include message in semaphore queue
  else  delete used process communication blocks;
        send message to process;
        start address:=branch address;
        place process in active queue;
        place current process in active queue;
        take next active process for execution;
```

61

```
end if;
  open interruption;
end;
```

Conclusion. Based on analysis of interaction of processes in a communications processor, we have presented in this work a communications mechanism that offers the following capabilities:
--processes can be synchronized by using signal variables or message transfer;
--required type of interaction is implemented by including appropriate data structures and procedure names in generalized semaphore control block; and
--combined waiting by a process for several events or resources is allowed.

Using generalized semaphores facilitates programming complex situations that occur in the interaction of processes in a communications processor. The capability of constructing enlarged and specialized communication facilities on the basis of generalized semaphores also reduces the overhead needed to afford process interaction.

Based on this mechanism for communication of processes in a communications processor control program, binary and general semaphores, semaphores with data, events and events with data have been implemented.

BIBLIOGRAPHY

1. Gusev, V. V.; Koval'chuk, V. K.; Kushner, E. F.; and Chernat, A. P., "Communication Processor Control Program for Computer Network with Packet Switching," USiM, No 6, 1979, pp 9-15.

2. Naumov, V. V.; Peledov, G. V.; Timofeyev, Yu. A.; and Chekalov, A. G., "Supervizor OS YeS EVM" [Operating System Supervisor for Unified System Computers], Moscow, Statistika, 1975, 88 pages.

3. "Programming Languages," ed. by F. Zhenyui, Moscow, Mir, 1972, 406 pages.

4. Dijkstra, E. E., "The Structure of the T.H.E. Multiprogramming System," COMM. ACM, Vol 11, No 5, 1968, pp 341-346.

5. Tsichritzis, D. and Bernstein, P., "Operating Systems," Moscow, Mir, 1977, 336 pages.

6. Davis, D. and Barber, D., "Communication Networks for Computers," Moscow, Mir, 1976, 680 pages.

7. Kotov, V. Ye., "On Parallel Languages," KIBERNETIKA, No 3, 1980, pp 1-12.

8. Hoare, C. A. R., "Monitors: An Operating System Structuring Concept," COMM. ACM, Vol 17, No 10, 1974, pp 549-557.

9. Hansen, P. B., "The Programming Language Concurrent PASCAL," IEEE TRANS. SOFTWARE ENG., Vol 1, No 2, 1975, pp 199-207.

10. Balzer, P. M., "Ports: A Method for Dynamic Interprogram Communication and Control," in "AFIPS Conf. Proc., SJCC," 1971, pp 485-489.

11. Dedkov, A. F. and Shchers, A. L., "Language for Development of Programs with

Distributed Structure in Computer Networks," PROBLEMY MSNTI/MTsNTI, No 2, 1981, pp 60-67.

12. Andrews, G. R., "The Design of a Message Switching System: An Application and Evaluation of MODULA," IEEE TRANS. SOFTWARE ENG., Vol 5, No 2, 1979, pp 138-147.

8545
CSO: 1863/37

INTEGRATED MULTIPROGRAMMING IN UNIFIED SYSTEM OPERATING SYSTEM

[Article by Filipp Illarionovich Andon, candidate of physicomathematical sciences;
Valentin Aleksandrovich Deretskiy, engineer; and Boris Yefimovich Polyachenko,
candidate of engineering science; all from the SKTB PO IK AN USSR [Special Process
Design Bureau, Software, UkSSR Academy of Sciences Institute of Cybernetics]
(Kiev)]

[Text] Problems of efficient operation of computer systems (VS) are becoming more
relevant with the emergence of complex and expensive computers. Solving them in
practice comes down to developing software components that extend the capabilities
of operating systems. As a rule, these are dispatchers and system meters for vari-
ous purposes [1-4], the basic functions of which are management of input job
streams and informing the user and computer center administration of the actual
load of computer system components in a given time interval.

Meter information is used in the strategy of selecting multiprogram mixes in subse-
quent stages of the computing process. Using such control systems in mixes of
various types of tasks yields a satisfactory result from the viewpoint of effi-
ciency in using computer system capabilities. For the class of data processing
tasks (ASU [automated control system], ASOD [automated data processing system],
information systems of various orientation and others), using existing systems for
control of the computing process is less efficient in connection with the large
volume of I/O operations, the presence of information links between tasks, and the
large volumes of sequentially processed information. In the best case, the compu-
ter system load reaches 30 to 40 percent of maximum.[5-7]. Achieving an efficient
load of about 70 to 80 percent in using the capabilities offered by just the base
operating system (OS MVT) alone is a very difficult task.

The software suggested in this work, in the view of the authors, and also based on
experimental studies, will allow solving a considerable number of problems aimed at
improving the level of efficient loading of computer systems for the class of data
processing problems.

The suggested discipline for calculations is based on the concept of the computa-
tional graph-scheme (VGS) [8] which contains a program of the computational process
for a specific multiprogramming mix of application programs. Control of execution
of programs fixed in the computational graph-scheme is based on exchange by elements

(by records, segments, sequences [kortezh]) between them of information through areas in main storage (OP) in the computing system. For this, each data set m_j in main storage is allocated a buffer, the minimum size of which is the size of a physical record for external (initial and resulting for the computational graph-scheme) data sets and a logical record for internal (temporary, intermediate for the computational graph-scheme) data sets [9]. Maximum size is limited to size of main storage for a specific computer system. Selecting the sizes of storage areas for buffers of data sets in the computational graph-scheme is a discrete programming problem [10-14], and the efficiency of the process as a whole largely depends on the solution to this problem.

In using integrated multiprogramming, the information in buffer m_j in main storage is used simultaneously by all programs that propose working with this data set.

To implement this organization of computations, the authors have developed software for Unified System computers operating in a YeS OS environment, versions 4.1 and 6.1. A general description and the main features of it follow below.

Let us demonstrate this type of organization of computations by an example. In figure 1, program r_2 produces data set m_4. It is the input for programs r_3 and r_4. The data set m_4 in main storage is allocated an area (buffer), to which all the programs (r_2, r_3, r_4) have access. Control consists in synchronizing the operation of programs r_2, r_3 and r_4 with set m_4 during processing of it by elements. Program operation is synchronized by the control program in the suggested software. All the application programs (programs r_1, r_2, r_3 and r_4 in fig. 1) are subtasks in relation to the control program. Software operation is based on intercepting application program calls for the I/O control system in YeS OS, analyzing them and substituting the SVC-programs inherent to integrated multiprogramming. In this case, the application programs may be written in any YeS OS language with use of standard exchange facilities. As a function of the

Fig. 1

status of the buffers allocated to the data sets in main storage, the application programs are given control or put in a wait state.
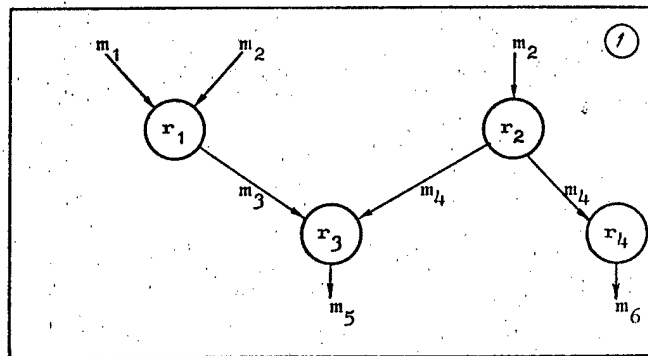
In the example in question, program r_2, generating elements of data set m_4, will fill the buffer allocated to this data set. After this, program r_2 will be put in a wait state until programs r_3 and r_4 have completely processed all data elements in buffer m_4. Then program r_2 is started etc. If program r_3 (r_4) makes a request for the next data element in set m_4, which has not yet been fomrulated by program r_2, program r_3 (r_4) is put in the wait state.

Let us call integrated multiprogramming this asynchronous multiprogram execution of tasks in which information-related programs process common data sets simultaneously.

Software for the System of Organization of the Process of Integrated Multiprogramming. Organization of control of integrated multiprogramming has affected a large number of components that support management of the computer process. The main ones are: task management, data management and resolution of impasse situations. The control unit in the proposed organization of computations is the computational graph-scheme, the contents of which reflect the status of the computing process. Synthesis of the computational graph-scheme in this version of the proposed software is performed up to implementation of it on the basis of descriptions of programs and data by facilities of the language suggested below. This solution meets the requirements of regular tasks in automated control and automated data processing systems and allows rather simply solving the problem of synthesis of the computational graph-scheme by using the macroassembler in YeS OS [15-16]. Development of an integrated multiprogramming system presupposes different alternatives for developing the computational graph-scheme, including developing one for random mixes of programs in real-time systems. Let us limit the discussion to software for the proposed integrated multiprogramming system.

The software has been described by two stages independent in time:
—design of the computational graph-scheme; and
—execution of it in the computing system.

A general diagram of the interaction between the integrated multiprogramming system and OS is shown in fig. 2.

Key:
1. computational graph-scheme synthesis stage
2. computational graph-scheme implementation stage
3. OS, Assembler, Linkage editor
4. computational graph-scheme synthesis facilities
5. computational graph-scheme G(R, M)
6. control information
7. data flow
8. computational graph-scheme
9. OS, Task initiator, Multiprogramming
10. integrated multiprogramming control
11. OS, Main storage control, Multiprogramming
12. application programs
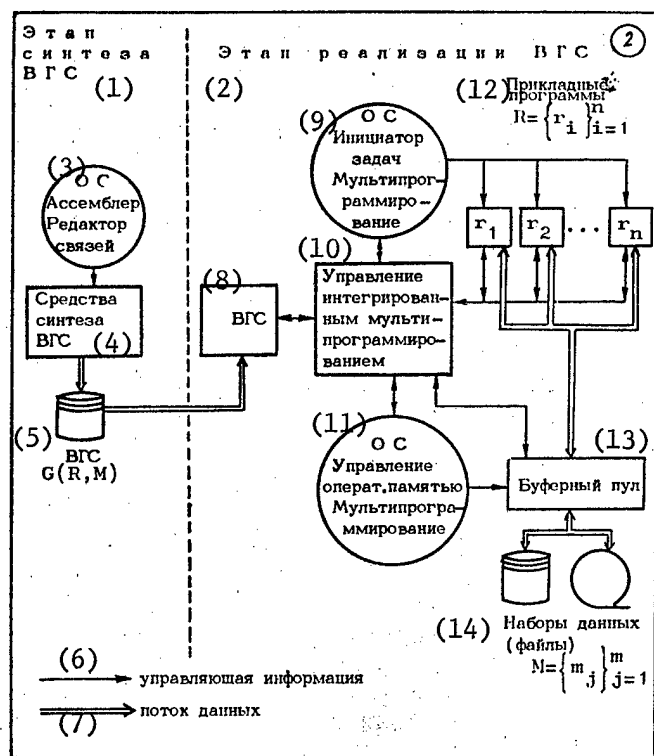13. buffer pool
14. data sets (files)

Fig. 2

In the first stage, the language suggested below is used to describe a set of programs, R. The programs R are a specific subsystem of the automated control and automated data processing systems. The result of execution this part of the work is the computational graph-scheme in a format known to the control programs in the integrated multiprogramming system. The computational graph-scheme is placed in a special data set.

In the second stage (see fig. 2), the requirement to execute the computational graph-scheme with a specific name is contained in the control stream of the job control language. The programs supporting the mode of integrated multiprogramming are loaded; they, in turn, load the indicated computational graph-scheme into main storage. Using the information in the computational graph-scheme, the control program initiates the application programs and controls the programs in the dynamics of their interaction in accordance with the data sets being processed. When the computing system resources do not permit placing in main storage all the programs and areas for buffers of the data sets (this information is known to the user at the computational graph-scheme synthesis stage) in the subsystem of the automated control and automated data processing systems, there is a facility that allows during design linking the independent computational graph-schemes in a chain of subschemes, each element of which meets the limitations on the computing system resources. Branching to execution of the next subscheme upon completion (normal or abnormal, but provided for under the condition of execution of the subscheme) of the current computational graph-scheme is effected under control of the integrated multiprogramming programs. Below, we shall dwell in more detail on the main aspects in design of the computational graph-scheme and some features of the algorithm for control of integrated execution of programs in the YeS OS environment.

Computational Graph-Scheme Design. The facilities suggested below are the tools for development of a computational graph-scheme, the extension of functions of which is considered an evolution of the integrated multiprogramming facilities.

The language for developing the computational graph-scheme is represented by a set of macroinstructions in YeS OS Assembler that provide for specification of characteristics of programs and data sets and formation of fields containing control information for programs mentioned in the computational graph-scheme. The computational graph-scheme, in a form ready for use, is identified and placed by the linkage editor in a data set indicated by the user. Four main macroinstructions have been developed to form the computational graph-scheme:
--one to describe tasks;
--one to form fields of control information for program r_i;
--one to describe a data set; and
--one to generate the computational graph-scheme.

For clarity in the computational graph-scheme synthesis process, let us consider the formats and description of the macroinstruction fields. The macroinstruction format for describing a task is as follows:

<name> DECL <name of module r_i>

FILEIN = <name of DD1 |, name of DD2 |, ...>,
FILEOUT = <name of DD1 |, name of DD2 |, ...>,
FILEON = <name of DD1 |, name of DD2 |, ...>

The macroinstruction parameter field contains one positional parameter: the name of the root absolute module of program r_i. The values of the key parameters are as follows:

FILEIN defines the list of DD-names of data sets input for the program;

FILEOUT defines the list of DD-names of data sets output for program r_i; and

FILEON indicates the list of DD-names of data sets that are recorded in one of two parameters presented above, but the data set has to be in external storage (tape, disk) or placed there.

The macroinstruction for describing fields of control information for program r_i is represented the following way:

$$\text{<name>} \ \_ \ \text{PARM} \ \_ \ \text{<text>} \ [,A=\begin{Bmatrix} S \\ L \end{Bmatrix}] .$$

Symbol or address parameters for the program, the description of which precedes this macroinstruction, can be sent to the macroinstruction field <text>. The key parameter A defines the type of information in the <text> field. When A=S, the <text> field contains a series of symbols which are sent to the program in accordance with the standard agreement on sending parameters in YeS OS. When A=L, the <text> field contains a series of address constants.

The macroinstruction format for describing a data set DCBDT is as follows:

<basic DD-name> DCBDT <key parameters of DCB>,

BFR= <number>,
DDNAME=<name of DD1 |, name of DD2 |, ...>,
TSKDD= <name of T1|, name of T2|, ...>,

$$ON= \begin{Bmatrix} R \\ W \end{Bmatrix} \ \_$$

The macroinstruction identifier is the DD-name replacing the DD-names, the synonyms from the DDNAME list. The DDNAME parameter is the list of all references of DD-names of tasks indicated in the computational graph-scheme. When references to DD-names have to be distinguished from specific programs, we use the TSKDD parameter, the series of elements in which must correspond to the series of elements in the DDNAME parameter.

The parameter BFR specified the number of blocks allocated in main storage as the buffer for data set m_j. The parameter ON indicates the type of external data set with respect tp the computational graph-scheme (input-output). The DCB key parameters provide the capability of indicating in the DCBDT macroinstruction all the parameters that can be specified in the DCB macroinstruction in YeS OS Assembler [16].

The final macroinstruction in the series of macroinstructions for generating the computational graph-scheme is the GRAPH macroinstruction:

<name> GRAPH NMG=<name of current computational graph-scheme>

[;NXTNAME=<name of next computational graph-scheme in chain>] ,

The NMG parameter indicates the name assigned to a given computational graph-scheme. NXTNAME parameters indicate the name of the computational graph-scheme that must be executed after completion of execution of the current one.

Let us discuss the technology of developing a computational graph-scheme for data processing systems by using the example of the BALANS task, developed in the Ukrsel'khoztekhsistem [System for the Ukrainian Agricultural Equipment Association].

The BALANS task is a standard subsystem in the ASU [management information system] for output of reports for specified time intervals by nomenclature status in production or in an industrial sector. The output data sets are a set of output reports for the different departments in an enterprise. For output of generalized reports, the combined input data set is sorted by the required features. The sorted data set becomes the main set for generating documents. The type of computational graph-scheme for the BALANS system is shown in figure 3.

The scheme contains five programs, {S1, S2, B2, B3 and B145}, and six main data sets processed sequentially, BLNS1, BM7, BM145, P1, P2 and P3. The other sets (non-sequential), IS1 and IS2, are processed by base system facilities.

Implementation of the Computational Graph-Scheme. At the stage of synthesizing the computational graph-scheme, there is no provision for making any changes to the existing debugged complex of applications programs in the management information and automated data processing systems. The changes needed to enable integrated access to data are made dynamically at the level of YeS OS access methods. Without going into the details that are inherent to the majority of control systems (initiation of programs, response to normal and abnormal ends, ensuring reliability of the computing

Fig. 3

Key:
a. data sets processed by standard OS facilities
b. data sets processed in integrated manner

process, etc.), let us just cover the elements that are peculiar only to the system in question and which distinguish it from existing systems for controlling the computing process. There are three such elements:
—communication with the operating system;
—task management; and
—I/O control.

Communication with OS is effected in two cases:
—recording of a password and list of addresses of modules for integrated access control in the CVTUSER field of the vector for OS communications; and
—integrated access program communication with the OPEN and CLOSE system programs.

Recording a password and lists of addresses of control programs is necessary to identify I/O requests from applications programs. Requests come from various sections of main storage. It is necessary to localize I/O requests (OPEN, CLOSE) coming from programs fixed in the computational graph-scheme that are executed in a specific main storage partition. Communication with the system programs IGC0001I
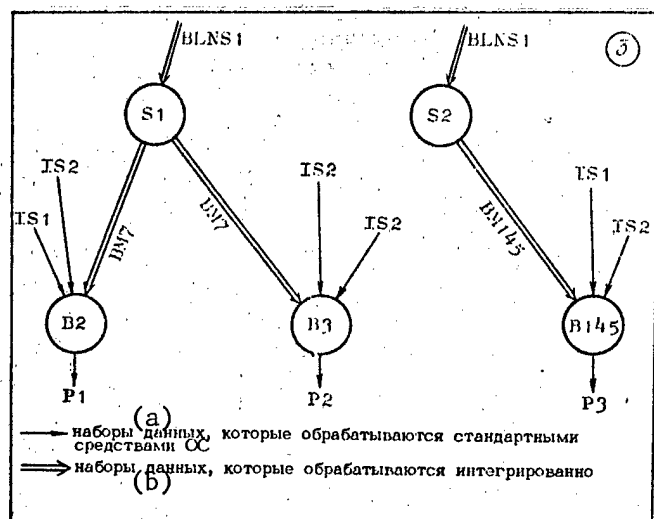
69

and IGC0002{ (OPEN, CLOSE) is effected in the control program for integrated access upon arrival of the appropriate requests from the applications programs in the computational graph-scheme. The essence of the communication consists in re-placing the instructions in the system modules for processing the OPEN and CLOSE macroinstructions with instructions for branching to the program for checking for the availability of integrated access.

If the request is not planned in the computational graph-scheme, the replaced in-structions are executed and control is returned to the standard branch for proces-sing the OPEN and CLOSE macroinstructions. When the request is scheduled, control passes to the programs for OPEN and CLOSE of integrated access. The CLOSE request causes program r_i to discontinue processing set m_j. The OPEN request causes switching to integrated access. The logic of multi-scanning of data set m_i by program r_i is constructed with regard to OPEN and CLOSE requests.

The program control algorithm is based on scanning the status of the computational graph-scheme for each request for exchange of programs recorded in the computa-tional graph-scheme. In main storage, the computational graph-scheme is repre-sented by a matrix of incidences of the graph $G(R, M)$, the columns of which cor-respond to data sets

$$M = \{m_j\}^m_{j=1},$$

and the rows to programs

$$R = \{r_i\}^n_{i=1}.$$

An element t_{ij} of the matrix (programs r_i and data sets m_j) fixes the current status of requests from a program for the corresponding data set. For each column, there is assigned an element s_j that defines the status of the buffer for an m_j data set in main storage. The basis for control (synchronization) of the programs is the result of comparing the statuses of requests of programs (t_{ij}) and statuses of buffers (s_j).

The diversity of existing access methods in the base OS and ensuring the insensi-tivity of application programs to changes in the organization of access to data impose considerable demands on an integrated multiprogramming system. Attention has also been paid to the efficiency of exchange of buffers of data sets m_i with external storage (MD, ML [disk, tape]) by using the apparatus of data chains [17].

Experimental Studies. To confirm the efficiency of the suggested system for con-trolling the computing process, experimental studies were performed; they demon-strated a considerable advantage of the organization in question for controlling the handling of data processing tasks. The throughput of large and medium computer systems increases two- to ten-fold compared to traditional control facilities. This increase in throughput when integrated multiporgramming facilities are used is due to full utilization of the central processor for tasks in the class in question. Use of central processor operation for control of the computing process under inte-grated multiporgramming is commensurate with use in the base multiprogramming sys-tem in handling data processing class tasks.

Results of measurements of the BALANS task executed on a YeS1060 in the modes of integrated and base multiprogramming are given below.

Parameter Measured	Multiprogramming Base	Integrated
CPU load	22%	89%
channel load	63%	48%
duration of task residence in main storage	24.5 minutes	7.75 minutes

Features that distinguish the suggested method from the base method of execution of the BALANS computational graph-scheme are:
—the data set BLNS1 will be read once;
—data sets BM7 and BM145 will not be placed in external storage when the computational graph-scheme is implemented; and
—the information-related tasks S1-B2, B3 and S2-B145 operate in a multiprogram manner.

Development, improvement and study of the software for the integrated multiprogramming system are continuing. Plans call for expanding the number of access methods connectable to the integrated multiprogramming system, the data processing modes and the means facilitating development of the computational graph-scheme and planning.

The software has been introduced in two organizations and has been sent to the Ukrainian Republic Library of Algorithms and Programs [18].

BIBLIOGRAPHY

1. Stogniy, A. A.; Pan'shin, B. N.; and Shafrov, A. Ya., "Software for Organizing the Computing Process in Management Information Systems: Current Status and Prospects for Development," VOPROSY RADIOELEKTRONIKI. SER. EVT, No 7, 1980, pp 13-22.

2. Kondrat'yeva, T. V.; Petrova, T. A.; Rokhlin, F. Z.; and Fishman, N. K., "KROS Subsystem of Planning," USiM, No 6, 1976, pp 26-29.

3. Pan'shin, B. N., "Software Package for Efficient Management of Computer Resources in Data Processing Systems," USiM, No 4, 1980, pp 127-130.

4. Pan'shin, B. N.: Myagi, A. R.; and Sume, V. E., "Software for Rating and Enhancing Efficiency of Use of Unified System Computers," Preprint 81-7, Institute of Cybernetics, UkSSR Academy of Sciences, Kiev, 1981, 52 pages.

5. Kachkov, V. P., "Rating Efficiency of Use of Input/Output Channels in Modern Computers," VOPROSY RADIOELEKTRONIKI, SER. EVT, No 12, 1971, pp 80-85.

6. Pogodin, A. Ye., "Ways of Improving Throughput of Computing Facilities," VOPROSY RADIOELEKTRONIKI. SER. EVT, No 8, 1980, pp 131-134.

7. Umbetbayeva, B. A., "Some Facilities for Design of Process Algorithms for Handling Tasks in Management Information Systems," USiM, No 5, 1980, pp 71-76.

8. Andon, F. I., "Posledovatel'nyye vychislitel'nyye graf-skhemy i ikh realizatsiya na EVM" [Sequential Computational Graph-Schemes and Implementation of Them on Computers], Preprint 77-74, Institute of Cybernetics, UkSSR Academy of Sciences, Kiev, 1977, 24 pages.

9. Glushkov, V. M.; Ignat'yev, M. B.; Myasnikov, V. A.; and Torgashev, V. A., "Rekursivnyye mashiny i vychislitel'naya tekhnika" [Recursive Machines and Computer Engineering], Preprint 74-57, Institute of Cybernetics, UkSSR Academy of Sciences, Kiev, 1974, 24 pages.

10. Andon, F. I. and Zhmurov, A. A., "Approach to Reducing Exchanges in Work with Files," KIBERNETIKA, No 6, 1974, pp 134-140.

11. Andon, F. I.; Kuksa, A. I.; and Polyachenko, B. Ye., "Efficient Computer Implementation of A Class of Informationally Related Programs," PROGRAMMIROVANIYE, No 3, 1980, pp 57-63.

12. Andon, F. I. and Polyachenko, B. Ye., "A Technology for Data Processing in Management Information Systems," KIBERNETIKA, No 2, 1980, pp 65-69.

13. Andon, F. I. and Polyachenko, B. Ye., "Problem of Optimal Distribution of Buffer Storage for Data Processing Programs," ELEKTRONNOYE MODELIROVANIYE, No 2, 1979, pp 108-111.

14. Polyachenko, B. Ye., "Solution to One Problem of Efficient Organization of Computing Process," ELEKTRONNOYE MODELIROVANIYE, No 1, 1979, pp 105-107.

15. Khusainov, B. S., "Makrosredstva v yazyke assemblera YeS EVM" [Macrofacilities in Assembler Language for Unified System of Computers], Moscow, Statistika, 1978, 94 pages.

16. Brich, Z. S.; Voyush, V. I.; Degtyareva, G. S.; and Kovalevich, E. V., "Programmirovaniye na yazyke assemblera YeS EVM" [Assembler Language Programming for the Unified System of Computers], Moscow, Statistika, 1975, 296 pages.

17. Larionov, A. M., ed., "Sistema matematicheskogo obespecheniya YeS EVM" [System of Software for Unified System of Computers], Moscow, Statistika, 1974, 144 pages.

18. Andon, F. I.; Dovgopolyy, V. G.; Kirik, E. G. et al.,"Programmnoye obespecheniye tekhnologii integrirovannogo mul'tiprogrammirovaniya" [Integrated Multiprogramming Technology Software], Kiev, 1981, 56 pages, RFAP [Republic Library of Algorithms and Programs] registration number 5957.

GENERATION OF PROGRAM CROSS FACILITIES FOR COMPUTER SYSTEM SIMULATION

[Article by Vladimir Aleksandrovich Smol'nikov, graduate student, Institute of
Cybernetics, UkSSR Academy of Sciences (Kiev)]

[Excerpts] A great deal of experience has now been gained in software simulation.
There are many simulation languages and systems (GPSS, SMPL, DDL, NYeDIS, ALSIM,
PROYeKT, YaUZA [3, 4]), but in practice, analysts and developers cannot always
make use of existing simulation systems because of the bulk and complexity in
investigating them or because of the lack of them in the standard software for
large computers. Therefore, complex systems have to be programmed each time
in general-purpose languages with large expenditures of resources.

Discussed in this work is the GENMOD (GENerator of MODeling Complexes) cross pro-
gram system developed by the author. It allows implementing partial automation in
developing simulation program systems (MPK) and facilitating operation of them by
users without system programming skills. This system also offers the user the
common language facilities as a tool for interaction with simulation program
systems.

Let us consider the main functions a simulation program system has to perform:
first, simulation of operation of the computer devices being designed at various
levels of detail and analysis of their functioning, and second, support for deve-
lopment and debugging of the program product already in the early stages of
computer system design.

The GENMOD system was implemented in a rather brief period in a YeS OS environment
by the "immersion" method [2] and allows making use of OS system support. There
are about 5,000 PL/1 statements in the GENMOD system.

In conclusion, one can note that the GENMOD system permits by user specification
developing the program modules of a computer system with which one can effect
analysis of the operation of computer systems being designed and debug the software.
A unified algorithmic language for interaction with the individual components in
simulation program systems has been developed in the system.

The GENMOD system was used in designing the Etalon specialized computer system
intended for a broad class of problems of approximate analysis of full-scale tests

[7]. The program model versions generated allowed on the basis of functional analysis refining the design solutions and optimizing the instruction set and structure. Also, more than 15,000 Assembler statements were used in developing the systems and applications software.

Key:
1. monitor
2. auxiliary subsystem
3. subsystem for debugging
4. interpreter
5. subsystem for analysis and special programs
6. clusters
7. system interface modules
8. system tables and objects

The adequacy of the program module, the powerful debugging subsystem in the GENMOD system and the broad set of service functions in the Unified System of Computers made it possible to completely perform the development and debugging of the software being designed for the Etalon computer complex in a cross version; several man-months were expended on this even before the base processer was manufactured.
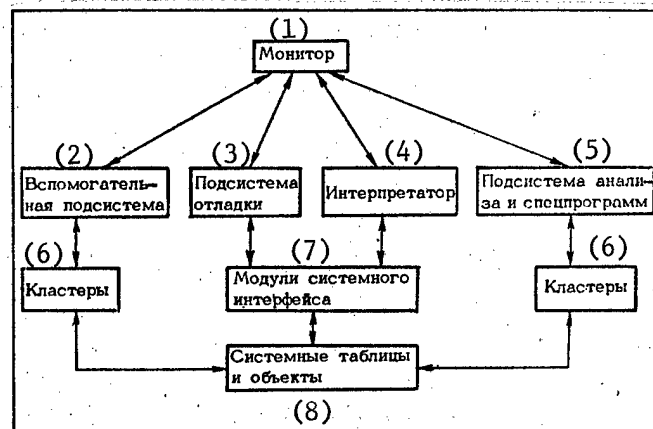
Fig. Structure of simulation program system generated and interaction of functional blocks

BIBLIOGRAPHY

1. Moralee, D., "Microprocessor Architectures: Ten Years of Development," ELECTRON. AND POWER, Vol 27, No 3, 1981, pp 214-221.

2. Nikitin, A. I., "Obshcheye programmnoye obespecheniye sistem real'nogo vremeni" [General Software for Real-Time Systems], Kiev, Naukova dumka, 1980, 134 pages.

3. Breyer, N., ed., "Computer-Aided Design of Computer Systems. Simulation Languages and Data Bases," Moscow, Mir, 1979, 462 pages.

4. Glushkov, V. M.; Kapitonova, Yu. V.; and Letichevskiy, A. A., "Avtomatizatsiya proyektirovaniya vychislitel'nykh mashin" [Computer-Aided Design of Computers], Kiev, Naukova dumka, 1975, 230 pages.

5. Byakov, A. Yu.; Gontarenko, S. D.; and Kuznetsova, S. D., "Klastery operatsionnykh sistem tsentral'nogo protsessora" [Central Processor Operating System Clusters], Moscow, 1979, 14 pages (ITM i VT AN SSSR [Institute of Precision Mechanics and Computer Engineering, USSR Academy of Sciences], Preprint 17).

6. Rastin, R., ed., "Large System Debugging Facilities," Moscow, Statistika, 1977, 134 pages.

7. Korniyenko, G. I., "Etalon Problem-Oriented Digital Computer for Real-Time Systems," USiM, No 1, 1979, pp 104-106.

8545
CSO: 1863/37

EXTENSION OF UNIFIED SYSTEM OPERATING SYSTEM R-TECHNOLOGICAL COMPLEX FACILITIES
IN IMPLEMENTING TECHNICAL AND ECONOMIC INFORMATION PROCESSING SYSTEMS

[Article by Lyubov' Nikolayevna Voinova and Viktor Vladimirovich Karpenko, both
engineers from the SKTB PO IK AN USSR [Special Process Design Bureau, Software,
Institute of Cybernetics, UkSSR Academy of Sciences] (Kiev)]

[Text] In recent years, R-technology of programming has become not just a subject
for discussion, study and bringing out the strong and weak points, but also a real
tool affecting the quality and quantity of the program product being produced.
This has also been confirmed by the four-year experience of the authors in develop-
ing software for technical and economic information processing systems by using
the YeS OS RTK technological complex.

Systems in this class have special query languages that specify the requirements
of the users of these systems for information processing. Considering that
queries in these languages can be rather complex expressions from the viewpoint of
both syntax and semantics, one can understand why it was R-technology that was
chosen as the means for developing software for these systems.

The experience gained in the process of these efforts showed all the advantages of
the R-technology of programming: more profound comprehensive automation of the
work of programmers, planning and management of groups of programmers, reduction
in the schedule for production and enhancement of the quality of the program pro-
duct, automation of the processes of documentation and maintenance of program
systems, etc.

The aim of this article is to investigate a major capability of the YeS OS RTK:
extension of the functional facilities of the technological complex, shaped by
the specifics of the tasks to be implemented.

Extension of the Capabilities of Accessing Data in YeS OS RTK. R/TRAN [1] is the
input language for YeS OS RTK. The main means of accessing data stored on exter-
nal media is the operation of connecting the USIN interface program for redefini-
tion of the input string. This program operates with data stored on external
media in the move mode, i.e. each successive logical record kept in the input buf-
fer is moved to a work area where it is processed [2]. When there is a small
volume of input data, use of the USIN program has no critical impact on processing
time.

But the use of relatively large input files entails considerable time for searching for information in them. This led to the need of developing the WLR program that effects input of information from files on magnetic disks in the pointing mode, in which a work area is not used; the address of a logical record in the input buffer is sent to the processing program.

The WLR program, written in YeS OS Assembler, performs the following:
—it opens the data control block (DCB) [2], associated with the data set to be read, in which the required file is kept;
—it reads the data set records into the input buffer;
—it sends the address of the record in the input buffer, which has been read, to the processing program written in the R/TRAN language; and
—it closes the DCB when the last record in the data set has been read.

The WLR program is in the form of a function; it generates a return code of "0" when the next data set record is read, or "1" when the last data set record is read and the DCB associated with this set is closed.

The program is called as follows:

WLR (FILE, REG),

where FILE is the name of the RTK file, with the use of which the data set is read (the name of the concrete physical data set must be specified prior to calling WLR by calling the subroutine for redefinition of file parameters, R#DEFIL [1]), and REG is the RTK register holding the information read from the data set.

Here is a program section that prints the data set EXAMPLE1 by using WLR:

```
    :
    :
  P1 * R # DEFIL(FILE1, 'DD1', 'EXAMPLE1')        P2
  P2 WLR(FILE1,REG1): P3 R # PRINTF<-(REG1,LF)     P2
  P3 * R # PRINTF<-('END OF PRINTOUT'/)            P4
```

Using WLR to read data sets containing several tens of thousands of records has allowed reducing processing time three- to four-fold compared to using USIN.

Additional Facilities for Processing Data on External Media. In a system developed by the authors of this article, the data base was a set of sequential data sets. File storage was provided for in YeS OS RTK to work with these data sets. In particular, with the use of the RTK file, one can effect a logical relation with the physically existing (or being created) YeS OS data sets. This relation permits making a record in the file and reading from it. But to update and/or delete data from the base, this proved inadequate. To perform these functions, three programs were developed in YeS OS Assembler: VRUP, PUTREC and CLOSEF, which update sequential data sets.

VRUP performs the following:
—it opens the DCB associated with the data set to be updated, in the mode of updating a sequential data set;
—it reads a data set record into the I/O area; and
—it sends the address of the I/O area holding the record to the program written in the R/TRAN language.

The VRUP program is in the form of a function; it generates a return code of "0" when the next data set record is read, or "1" when the last data set record is read. The program is called like this:

VRUP (FILE, REG, ADR),

where FILE is the name of the RTK file, with the use of which the data set is read; REG is the RTK register holding the information read; and ADR is a 4-byte field containing the address of the I/O buffer holding the record to be updated.

The PUTREC program puts the updated information into the I/O buffer and writes it to the same place from which it was read by the VRUP program.

The CLOSEF program effects physical closing of the data set.

PUTREC and CLOSEF are subroutines without parameters and are called by name.

To illustrate the use of the VRUP, PUTREC and CLOSEF routines, here is an example of a program section that reads the sequential data set EXAMPLE2 and when the first two characters in the next record are AB, it replaces them with CD:

```
     .
     .
     .
  P1 * R # DEFIL(FILE2,'DD2','EXAMPLE2'),NUL(RABREG)                P2
  P2 VRUP(FILE2, REG2, ADR1): P4 R#SUBSTR (RABREG, 0, 2, REG2)      P3
  P3 R#EQRG (RABREG, 'AB': P2 R#INSTR (REG2, 0, 2, 'CD'), PUTREC    P2
  P4 * CLOSEF                                                       P5
     .
     .
     .
```

It should be noted that just as in WLR, the use of user programs written in Assembler entails no rigid restrictions on their structure and execution is standard as provided for in the R/TRAN language.

Additional Capabilities for Dynamic Loading of Programs. In developing large program complexes, it is necessary to solve the problem of creating modules with a dynamic structure, when there is in storage only a root subroutine that dynamically loads required module segments into storage. There are programs such as R#LINK, R#LOAD and R#DELETE in the R/TRAN language to organize the dynamic structure of a program module.

R#LINK loads into main storage the required module, executes it and then deletes it. R#LOAD loads into storage the required module and reports the address for the point of entry to it. R#DELETE deletes a specified module from main storage. But simple use of these facilities is possible only in the case when the calling and called segments have communication on data stored in external storage. But this approach is not possible when it is necessary to effect interprogram communication on data kept in main storage (recall that provided for in RTK are such types of memories as counters, registers, stacks, wagons, etc.).

To allow communication between segments on data kept in main storage in a module with dynamic structure, several macrodefinitions were developed. Let us consider the functions and conditions for using these macrodefinitions.

Key:

1. description of RTK memories, GERM
   registers: REG1, REG2, REG3
   counters: CT1, CT2
   stack: ST1
2. external program, AB GERM MODE=DEF1
3. root segment, AB GERM MODE=DEF
4. load segment, AB GERM MODE=USE
5. creation and processing
6. processing
7. RTK memories REG1, REG2, REG3, CT1, CT2, ST1
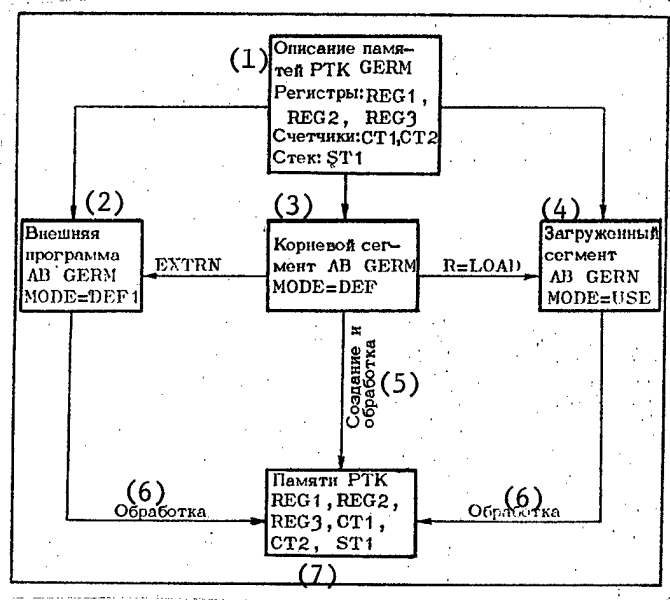
Example of Dynamic Loading

The macrodefinitions CT, RG, RS and RSV allow defining a counter, register, stack and wagon, respectively. Then the macrodefinition is created in which the names and dimensions of the memories by which communication between segments must be effected are specified by using the macrodefinitions listed above. Implemented in each segment is the macrocall of this macrodefinition in one of three modes provided: DEF, DEF1 and USE.

A macrodefinition with the mode DEF is specified in the root segment of the module with dynamic structure. In this case, space is allocated in main storage and all necessary parameters are specified for the required counters, registers, stacks and/or wagons.

A macrodefinition with the mode DEF1 is specified in the load segment of a module with dynamic structure. This mode allows a load module to operate with memories created in the root segment and makes the results of its operation received in the corresponding memories accessible to the root segment or other load modules.

A macrodefinition with the mode USE is used in programs which are connected to the root or called segments by using the statement for description of external names, EXTRN.

Operation with these macrofacilities is shown in the figure. The macrodefinition GERM describes the counters CT1 and CT2, registers REG1, REG2 and REG3, and the stack ST1. A macrocall of this macrodefinition with the name AB is implemented in the root, load and external segments.

The macrofacilities developed are stored in the RTK.MACLIB library in the RTK complex. Using them allows development of modules with a complex dynamic structure and avoiding, in doing so, superfluous exchanges with external storage units.

BIBLIOGRAPHY

1. Vel'bitskiy, I. V.; Khodakovskiy, V. N.; and Sholmov, L. I., "Tekhnologicheskiy kompleks proizvodstva programm na mashinakh YeS EVM i BESM-6" [Process System for Generation of Programs on Unified System and BESM-6 Computers], Moscow, Statistika, 1980, 262 pages.

2. Khusainov, B. S., "Programmirovaniye vvoda-vyvoda v OS YeS EVM na yazyke assemblera" [Input/Output Programming in Unified System OS in the Assembler Language], Moscow, Statistika, 1980, 264 pages.

8545
CSO: 1863/37

OPERATING SYSTEM FOR THE PAROM MICROCOMPUTER HOMOGENEOUS COMPUTER SYSTEM

[Article by Svetlana Al'bertovna Zuykova, Anatoliy Petrovich Marushchak, Vladimir Georgiyevich Mirin, Anatoliy Aleksandrovich Tumanov and Nikolay Arkad'yevich Khvostantsev; all engineers from the Smolensk Branch of the VGPTI TsSU SSSR [All-Union State Technological Design Institute, Central Statistical Administration, USSR] (Smolensk)]

[Text] The PAROM microcomputer homogeneous computer system (MOVS) is intended for building a large number of institutional systems oriented to handling economic accounting and information tasks. PAROM is a concentrated system that may include from 1 to 104 element machines (EM) in the Elektronika-60 family.

The system is designed on the basis of the concept of a common system channel to which the element machines are connected in parallel through their slices of the system processor. For efficient functioning, the translation type of intermachine exchange and the synchronization operation have been implemented by hardware in the system. To eliminate conflicts during individual (paired) interactions between the machines through the system channel, which in this case is a critical resource, a system channel binary semaphore has been implemented by hardware. All peripherals are connected to the individual channels of the element machines [1].

In institutional systems, a group of users, including planners, accountants, managers and others, interacts with the computer system in real time directly from terminals installed at work stations. Implementing the mode of real time with multiterminal access to the computer reources in the microcomputer homogeneous computer system imposes high demands on the efficiency of the operating system (OS), independence of the number of element machines, its reliability and support of necessary service functions.

Analysis of the algorithms for the tasks handled by institutional systems allows identifying three classes of programs that are objects of control for the MOVS OS. The first and broadest class includes sequential programs requiring no more than the resources of one computer for execution of them. The second and third classes include the parallel programs (P-programs). They are distinguished by nature of interaction of branches and the synchronization and exchange facilities used.

Parallel programs for solving weakly related problems usually consist of a small number of branches between which there is, as a rule, paired (asynchronous) interaction. Such programs include those for sending messages between terminals and those for access to remote files. Programming for these problems involves mainly using statements of individual interactions.

Paralell programs for solving strongly related problems [2, 3] are a unified object consisting of an organizational set of branches, each of which is executed in a separate element machine. Such programs may have a large and variable number of branches. Interaction between branches is effected mainly by using statements for group (synchronous) interactions [2, 3]. Some examples of such problems: retrieving information in large data bases, and large-scale plan and accounting problems solved on the basis of mathematical economic methods.

In turn, the MOVS OS is also a rather complex parallel program complex which in functioning makes use of both group and individual system interactions. The main problems in designing the MOVS OS are those of efficient management of the various classes of programs in on-line operation under the conditions of limits to the resources of the individual element machines in the system. Developing the MOVS OS on the basis of existing OS for the base Elektronika-60 microcomputer is reasonable for the initial versions, while for industrial systems optimized for the classes of problems to be solved, it is necessary to design problem-oriented OS that afford the required efficiency and cost.

MOVS OS Design Concept. Developing a general-purpose OS under the conditions of the limitedness of element machine resources and high demands on response time is not very efficient for the MOVS. The basis for designing the MOVS OS is the concept of a multilevel problem-oriented OS in which two major levels are identified: the base and the problem-oriented.

The base level in the MOVS OS, in turn, consists of several sublevels:
--supervisor of intermachine intervctions;
--system of standard functions;
--supervisor of storage;
--facilities for self-diagnostics and reconfiguration;
--I/O control system;
--facilities for interactive control of the system; and
--facilities for supporting the operation of the MOVS in a distributed homogeneous computer system.

The base level of the OS is a superstructure over the hardware and implements the basis of system operations of intermachine interactions, the physical I/O control system and other facilities supporting the design of various problem-oriented levels in the OS. Each class of application problems and modes of functioning gets its own problem-oriented level in the OS.

Problem orientation of an OS level consists in developing control system components such as specialized schedulers, access methods and interactive facilities for operating in real time.

Each sublevel of the MOVS OS base level can conceptually be considered a virtual homogeneous computer system with a specific set of functions. An upper sublevel can make use of the functions in sublevels beneath it. The functions of each sublevel are implemented by the corresponding program modules [4, 5].

OS servicing of request operations from a P-program is reduced to through access to the appropriate sublevel capable of implementing the function requested. Transfer of control from a user P-program to the MOVS OS goes in the top-down direction. Calls from an upper sublevel to lower ones may be used in implementing some functions within the OS.

Certain system procedures (individual interactions, start of a P-program, abnormal end of a P-program, reconfiguration when there are malfunctions in the MOVS) are asynchronous and are implemented by using a generalized unconditional branch (OBP) opervtion. It consists in simultaneous external interruption of all branches in the P-program and a call from the bottom up from the hardware to OS levels.

As a function of user requirements (specific composition of external units, need for availability of the mode for highly reliable computations, etc.), some MOVS OS levels may not be included in the specific version generated.

Separation of the major levels, the base and problem-oriented, in the makeup of the MOVS OS is the basis of the technological flexibility and expandability that promotes the most efficient design of sepcial control facilities in the various specific versions of the MOVS OS.
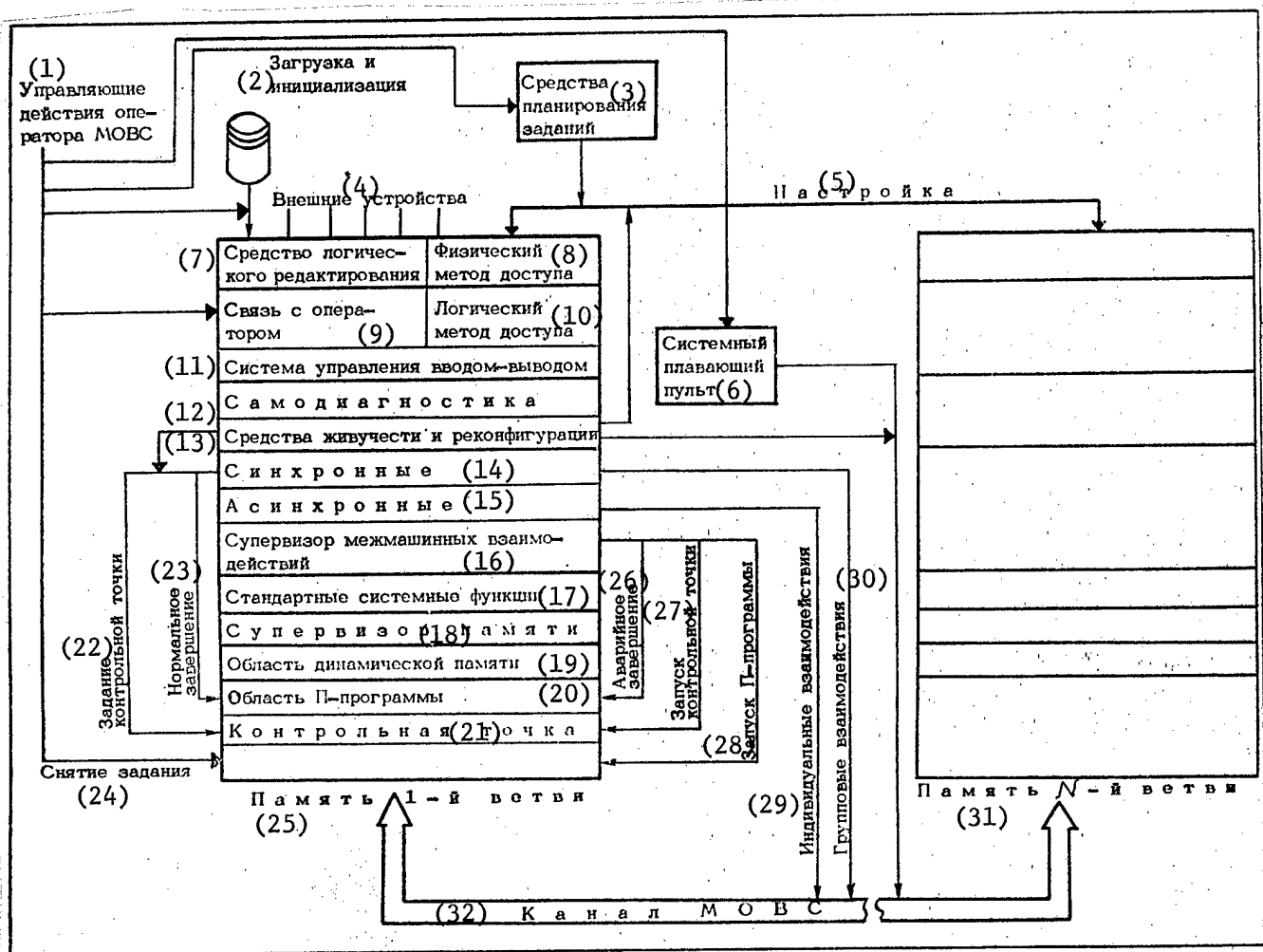
MOVS OS efficiency is due to the following factors:
--generation of a minimal set of base facilities with the problem-oriented facilities required in the specific application, keeping to the minimum thereby the storage taken up by the MOVS OS;
--design of the MOVS OS in the form of functional levels that promote optimization of time for meeting system requests because of through calling of the request to the appropriate functional level;
--considerable reduction of the complexity of control in each element machine owing to the lack of the multiprogram situation; and
--distribution of peripherals by individual element machines that promotes substantial reduction in the intensity of the stream of interruptions to be processed.

Enhanced protection of the MOVS OS against programmer errors is achieved by the chosen technological method of design [6, 7] in which decomposition of the MOVS OS extends to functional levels, the MOVS OS modules have informational reliability and intermodular communications are regular in nature.

This allows detecting an error situation in an individual element machine and taking measures for its elimination by other element machines; the error is localized to the functional level where it showed up and the error does not extend to other levels.

The MOVS OS has facilities to overcome deadlocks in using intermachine interactions in P-program branches executed concurrently. In this case, synchronous interactions are executed in the MOVS OS in accordance with the planning that occurs in each branch of a P-program, and the use of the common resource, the MOVS channel,

Diagram labels (Russian):

(1) Управляющие действия оператора МОВС
(2) Загрузка и инициализация
(3) Средства планирования заданий
(4) Внешние устройства
(5) Настройка
(6) Системный плавающий пульт
(7) Средство логического редактирования
(8) Физический метод доступа
(9) Связь с оператором
(10) Логический метод доступа
(11) Система управления вводом-выводом
(12) Самодиагностика
(13) Средства живучести и реконфигурации
(14) Синхронные
(15) Асинхронные
(16) Супервизор межмашинных взаимодействий
(17) Стандартные системные функции
(18) Супервизор памяти
(19) Область динамической памяти
(20) Область П-программы
(21) Контрольная точка
(22) Задание контрольной точки
(23) Нормальное завершение
(24) Снятие задания
(25) Память 1-й ветви
(26) Аварийное завершение
(27) Запуск контрольной точки
(28) Запуск П-программы
(29) Индивидуальные взаимодействия
(30) Групповые взаимодействия
(31) Память N-й ветви
(32) Канал МОВС

OS interaction with P-program, operator and external devices

Key:

1.	control actions of MOVS operator	17.	standard system functions
2.	loading and initiation	18.	storage supervisor
3.	job planning facilities	19.	dynamic memory area
4.	external devices	20.	P-program area
5.	set up	21.	checkpoint
6.	system floating console	22.	checkpoint job
7.	logic editing facility	23.	normal end
8.	physical access method	24.	job removal
9.	communication with operator	25.	storage of 1st branch
10.	logical access method	26.	abnormal end
11.	I/O control system	27.	checkpoint start
12.	self-diagnostics	28.	P-program start
13.	facilities for viability and reconfiguration	29.	individual interactions
14.	synchronous	30.	group interactions
15.	asynchronous	31.	storage of N-th branch
16.	intermachine interaction supervisor	32.	MOVS channel

in this case occurs without deadlock. Semaphore variables are not used in synchronous or group interactions, and all control of such interactions is based on specific standard events [8].

Use of asynchronous or individual interactions based on operations with a semaphore variable, which corresponds to the MOVS system channel as a shared resource, does not lead to a deadlock since semaphore operations are supported by a hardware facility for resolving conflicts in accessing the MOVS system channel [1] shared by P-program branches or weakly related programs.

OS Functional Structure. This OS has the following components:
--intermachine interaction supervisor;
--job planning facilities;
--I/O control system;
--storage supervisor;
--interactive system control facilities;
--OS simulator; [as published, should read "initiator"]
--self-diagnostic and reconfiguration facilities;
--system of standard functions;
--and facilities for supporting operation in a distributed homogeneous computer system.

Intermachine Interaction Supervisor. The OS nucleus is the intermachine interaction supervisor that supports data exchange between P-program branches executed concurrently, and starting and ending P-programs.

The basic type of interaction is synchronous, previously planned in all branches and is algorithmically determined. Synchronous interactions include the operation of generating a generalized condition for branching and operations for group exchanges of information.

A generalized conditional branch (OUP) consists in the generation by all branches in a P-program of a generalized condition that is a conjunction of the partial conditions specified by the individual branches. Using the valut of a generalized condition, all branches in a P-program can synchronously execute branching.

The set of group exchanges implements the basic alternatives for synchronous transfer of information in the MOVS.

In a translation exchange, one branch transfers a portion of the information, and the others receive it. A translation-cyclic exchange consists of several translation exchanges, executed alternately by each branch. As a result, accumulated in each branch are identical sets of portions of information from all the branches. In a differentiated exchange, there is one sending and several receiving branches. In view of the common synchronous nature of group exchanges, the branches not explicitly defined as receiving, effect a dummy reception. In the collector exchange, all branches send portions of data, and they are received by one branch. Conveyor exchange consists in sending portions of information from each branch to the adjacent branch with a higher number. From the last branch, the data is sent to the first.

Synchronous operations also include the operation of the normal end of a P-program in all branches, after which a new cycle of job planning begins, and the operation of determining a checkpoint from which a P-program can be restarted after one of the element machines in the MOVS has malfunctioned.

Asynchronous interactions are auxiliary; they are not used as a basic facility. They differ by the fact that two processes, active and passive, participate in them at any time. An active process represents a branch which is able to engage the shared system resource, the MOVS channel, and which determines a passive process in one of the other branches.

A facility for organizing asynchronous interactions is the operation of a generalized unconditional branch. It is initiated by a branch that has engaged the MOVS channel and consists in interrupting execution of programs in all branches of a P-program.

In OS, asynchronous interactions are used to start P-programs from an entry point or previously determined checkpoint in the mode of stable computations, and abnormal end of a P-program at the initiative of one of the branches, and to organize individual interactions between branches in a P-program.

In the mode of individual interactions, an active process may:
--write information to a passive process area;
--read information from a passive process area;
--specify a point from which a branch of a passive process restarts execution after the end of a session of an individual interaction; and
--force the halt of a program of a passive process branch.

Job Planning Facilities. The OS job scheduler is executed in the user program area. Its operation precedes the start of any P-program, at the end of which the OS automatically initiates loading and execution of the job scheduler. The main principle of design for job planning facilities in OS is supporting the capability of flexible, efficient job management for a broad class of applications. In accordance with this, the MOVS OS offers the user the following programs:
--BATCH SCHEDULER, which determines the characteristics for the next job in a dialog with the operator controlling the flow of batch jobs in the system;
--INTERACTIVE SCHEDULER, intended for MOVS operation within a distributed homogeneous computing system and which offers the non-specialist, computer system users the capability of starting their own jobs independently from remote terminals;
--and the USER SCHEDULER, which can be written for a special mode of starting jobs that is suited to a specific operational application.

I/O Control System. Any MOVS element machine can have connected to it such units as magnetic disk storage units (floppy and hard), magnetic tape storage units, mosaic printers and terminals.

The OS physical access method ensures in controlling I/O execution of read/write operations, special functions, system handling of I/O errors and handling of these errors by user programs.

The logical access method supports user operation with a terminal and offers the user the basic facilities for editing the information input from the keyboard.

Storage Supervisor. A P-program can use a free area of storage by using dynamic requests. The storage supervisor performs the following functions: allocation of storage sectors upon requests from a P-program, release of previously allocated sectors and checking for retention of an OS area.

Interactive System Control Facilities. The main functions of a MOVS computer installation operator include control of OS loading and initialization in the interactive mode and control of the mode of operation of the job planning facilities.

MOVS OS also gives the operator the capability of asynchronous, abnormal removal of an executing job, operating with the diagnostic system facility, the floating console, and the mode of asynchronous communication with a running P-program.

Operating System Initiator. The operation of the initiator when loading the OS consists in automatic definition of the MOVS configuration, loading and starting the MOVS OS in all the element machines and setting up the control blocks.

Self-Diagnostic and Reconfiguration Facilities. These facilities are based on the operation of self-diagnostics which permits timely and efficient detection of possible malfunctions that occur in the system and initiation of measures to restore good working order.

When an element machine goes down, the system executes the reconfiguration procedure, i.e. it removes the failed machine from the MOVS composition, changes the control blocks and continues operation. This is how the mode of highly reliable computations or the mode of MOVS viability are implemented.

In the mode of highly reliable computations, the P-program must ensure retention of information. After the failure of an element machine, to continue running a P-program, a checkpoint has to be determined in advance in the program to restart the P-program after reconfiguration of the MOVS.

The system diagnostic facility (floating console) provides the capability of implementing instructions in the "console terminal" mode for an Elektronika-60 computer alternately in any of the MOVS element machines. After loading the P-program, OS gives the operator the capability of inspecting and changing the program codes in any element machine before starting the P-program. Similar actions are provided for after an abnormal end of a P-program.

System of Standard Functions. To control the logic of P-programs and ensure parallel processing of data, OS allows a P-program to request its own number of a branch and the number of branches of the P-program and elementsassigned to this or any branch with the indicated number when there is uniform distribution, between branches of a P-program, of an array of N elements, as well as the ordinal number of the first element assigned to this or any branch with the indicated number after uniform distribution between branches of an array of N elements.

Facilities for Supporting Operation in a Distributed Homogeneous Computer System. When the MOVS is the node of a distributed or local-distributed system intended

for handling weakly related tasks (with rare interactions between branches), communication between nodes is organized at the job control level and facilities for control of protocols for exchange with remote terminals are used.

The version of MOVS OS now implemented for several institutional tasks has the following characteristics:

program rate	50K bytes/s
average size of module	68 lines of source text
storage taken up by OS	24K bytes
storage taken up by OS nucleus	12K bytes

BIBLIOGRAPHY

1. Gorin, V. G.; Indenbaum, V. M.; Korolev, V. M.; et al., "The PAROM Microcomputer Homogeneous Computer System," USiM, No 4, 1982, pp 66-70.

2. Yebreinov, E. V. and Kosarev, Yu. G., "Odnorodnyye universal'nyye vychislitel'nyye mashiny vysokoy proizvoditel'nosti" [Homogeneous, General-Purpose, High-Throughput Computers], Novosibirsk, SO [Siberian Branch] of Nauka, 1966, 308 p.

3. Yevreinov, E. V. and Khoroshevskiy, V. G., "Odnorodnyye vychislitel'nyye sistemy" [Homogeneous Computer Systems], Novosibirsk, SO Nauka, 1978, 320 p.

4. Yodan, E., "Structural Planning and Design of Programs," Moscow, Mir, 1979, 416 pages.

5. Hughes, J. and Michtom, J., "Structured Approach to Programming," Moscow, Mir, 1980, 278 pages.

6. Tsichritzis, D. and Bernstein, P., "Operating Systems," Moscow, Mir, 1977, 336 pages.

7. Mednik, S nad Donovan, J., "Operating Systems," Moscow, Mir, 1979, 628 pages.

8. Mirenkov, N. N., "Sistemnoye parallel'noye programmirovaniye" [Systems Parallel Programming], Novosibirsk, 1978, 50 pages (VTs SO AN SSSR [Computer Center, Siberian Branch, USSR Academy of Sciences], Preprint OVS-06).

APPLICATIONS

MORE EFFICIENT USE OF COMPUTERS IN ESTONIAN SSR PROPOSED

Tallinn SOVETSKAYA ESTONIYA in Russian 6 Jan 83 p 2

[Article by V. Alad'yev, chief of the Department of the Computer Center, ERK
[Estonian Republic Office] of the USSR Gosbank, and member of the International
Association for Mathematical Modeling: "The Effect We Have Been Waiting For: On
the Utilization of Computer Hardware in the Republic"]

[Excerpt] In Tallinn, there are now about 70 computer centers solving the various
problems of the national economy. The total overhead for operating them is rather
high, and the effect from the operation at times leaves much to be desired. What
is the reason for this?

First, with so many computer centers and the shortage of labor, especially highly
skilled, on the average, each of the centers does not have enough specialists with
the necessary skills (sometimes there is even a severe shortage). This is having
a serious impact on the efficiency of utilization of the available computer hard-
ware.

Further: the necessity of concentrating this hardware stands in contradiction to
the scattering of it in the numerous computer centers, a number of which have been
established without adequate economic and technical substantiation. Actually,
according to the data of the Estonian SSR Central Statistical Administration, --
and SOVETSKAYA ESTONIYA has written about this in detail already--a number of com-
puter centers in Tallinn not only do not maintain the required load on their own
hardware, but even achieve this very load often because of outside customers, by
renting machine time to them. These facts, of course, indicate that setting up
such centers was justified not by strict economic estimation, but rather by
voluntaristic decisions. Scattering of computer hardware leads to significant
duplication of effort, haphazard work, increased costs for information processing,
overdrawing on the reserves of skilled specialists, costs for maintenance, etc.

In the country, setting up multiuser computer centers and cluster computer centers
for the individual sectors and departments has become the means of raising the
efficiency of computer utilization in the national economy. (Among them, there is
one multiuser computer center in Tallinn.) The experience of operating them has
shown that concentration allows pursuing a common policy in equipping and reequip-
ping the computer centers, putting software development and the entire process of
information processing on an industrial basis, and in doing so, making use of the
most modern computer facilities that, as is well known, are beyond the resources

of the small computer center, etc. Moreover, concentration allows with less loss implementing the main task defined for computer hardware by the 26th CPSU Congress: development of computer /networks/ [boldface] and computer centers. The nodes of such networks must be major centers with industrial methods for information processing.

8545
CSO: 1863/61

STEPS SUGGESTED FOR IMPROVING COMPUTER USE IN KIEV

Moscow PRAVDA in Russian 27 Sep 82 p 3

[Article by O. Gusev, PRAVDA correspondent: "The Computer is a Collective
Machine"]

[Text] Kiev--"Nowadays, being without a computer is like not having hands"
is a statement that can frequently be heard in large plants, factories and
institutes. And really, electronics is increasingly lending us a hand. For
example in the Ukraine the first phase of a republic-wide automated control
system (RASU) recently began operating. And intensive work is now in progress
to set up the second phase of the system in the current five-year plan. The
RASU has been called upon to unify the capacities of hundreds of computing
centers and control systems on various levels in Kiev, Kharkov, Dnepropetrovsk,
Donetsk, Odessa and other cities of the Ukarine.

A large percentage of computers in the Ukrainian SSR are concentrated in the
capital of the republic. In just the last few years, the number of computing
centers in the city has doubled, and there has also been an increase in the
number of useful hours produced by each computer. Today there are computers
in enterprises, in transport, communications, science and educational organi-
zations. The practical problems involved in management are more and more being
solved with the aid of computers.

A dispatcher service has been set up at the city statistical administration
to take better advantage of these capabilities. The task of this service is
accounting for and maximally utilizing the reserves of all computers in Kiev.
And they are not a few -- the requirement for machine time in various organi-
zations changes at times not only in the course of a quarter or a month, but
even a day. Therefore such a service, knowing the needs for machine time of
other organizations at a certain hour, rapidly assists in finding the user
most in need of computer services.

There is no denying that the project has far-reaching goals: after all, in
Kiev as in other cities there are still a lot of enterprises that do not have
access to computer technology. Therefore from morning to night the dispatcher
service is flooded with order letters, telegrams, telephone calls: "urgently
need any computer available". The first efforts of the new service are already
noteworthy: For example in just the last year the dispatchers have discovered

and assisted with the sale of unoccupied computer time valued at several hundred thousand rubles.

And all the same, idle computers are still no rarity. For example several dozen computers were in operation for six months in Kiev last year without any definite plan. This means that computers somewhere were just spinning their wheels.

The causes? Mainly in burequcratic aloofness, and also in the lack of economic levers to stimulate the most complete utilization of reserves in many computing centers located in budget financing.

Analysis has shown that the above-mentioned computers have given such a high total output of machine-hours that it has automatically raised the general index by more than three percent. This has happened because the plan indices for some computers were understated relative to the normative values -- the machines have been operating with a much lower load than they could have: as strange as it may seem, some computer subdivisions to this day are unacquainted with normative documents on computer operation. And some republic organizations, including the ministries of State farms, land reclamation and water management, commerce, are poorly monitoring execution of their own subordinate work plans and normatives of computer loading. The UkSSR Ministry of the Construction Industry, for example, having acquired a considerable amount of modern computer equipment, is not making adequate use of it: among other things, for several years the ministry has not met the quota for loading machines with the highest productivity -- those of the third generation. Even gross violations in handling accounting and bookkeeping documentation have been noted and taken into consideration in the UkSSR State Planning Commission. When at a meeting of the commission of Kiev Party Committee that is charged with studying ways to improve efficiency of utilization of computer equipment an investigation was made into the possibilities for better utilization of computers on the scale of the capital city, it was found that merely eliminating city-wide losses of machine time through organizational-business causes would be equivalent to the acquisition of more than thirty computers -- much more than the annual increase in the inventory of such equipment in Kiev.

And just how are these reserves being utilized? To put it briefly -- unevenly. A few years ago, Kiev Central Department Store acquired some computers. The machines are working properly, helping do a considerable volume of work on accounting and planning sales operations. But complete efficiency is still being put off -- the information-computing center of the store has not been meeting the quota for computer loading since 1979 under the pretext of store repairs.

Another association of the city, "Analitpribor", has computer capabilities in excess of its own needs. However, it is not renting computer time to other organizations as the computer service group is understaffed. There is likewise a shortage of personnel at the republic-wide "Avtoremont" Association's computer information center, and the skill level of the present service staff is only slowly increasing.

These and other "illnesses" are not limited to the organizations mentioned above. Such deficiencies as inadequate operational-calendar planning and prolonged times of making computers operational, often exceeding the normative times, are typical of many other organizations as well. This situation is aggravated by the slow conversion of computer subdivisions to the quite profitable system of cost accounting with independent balance -- in the capital city of the Ukraine less than a fourth of the computers are cost-accounting, and unfortunately their number is increasing slowly.

A. F. Nezabitovskiy, director general of the Kiev production association "Elektronmash", confirms that far from all computers produced here are continuously loaded in full measure, and notes that from the very first stages of computer introduction, provisions should be made for favorable conditions on all stages of use. However, in practice this often is not the case; after acquiring a computer, clients delay putting it into operation for many weeks, and even months. Investigation of a number of computing centers in Kiev has shown that in nearly a quarter of the enterprises, a third of the scientific research institutes and more than half of the institutions of higher education, the equipment has been incomplete, and there have been delays in introducing certain components of calculating and problem-solving systems.

What should be done to overcome deficiencies of this kind? Specialists assure us that the first step should be taken even before considering requisitions for purchase of a computer. Here consideration must be taken not only of the production need, but also the preparedness of the enterprise or ministry for utilizing the machine on all stages from the instant of arrival at the railroad station. Experience has shown that only with a rigorous approach to computer distribution can we expect efficaceous use of a computer, and the efficiency of its application to improve at an accelerated pace with each stage of introduction.

Also very important is the widest possible use of computers for solving wide-scale management problems based on optimization methods of mathematical economics. Well-developed dialog forms of interaction of management specialists with computer systems (the "Displan" system developed under the direction of Academician V. M. Glushkov might serve as an example) considerably improves the quality of managerial decisions, and makes them more soundly based.

An inestimable contribution has been made here by the experimental collective-user computing center first set up in the Ukraine at the instigation of specialists of the Institute of Cybernetics, UkSSR Academy of Sciences in one of the displays at the All-Union Exhibition of Achievements of the National Economy of the republic. This center has also given an appreciable economic effect.

The "peak" mode of computer loading caused by the exigencies of production is typical even of such large computing centers as that belonging to the Institute of Cybernetics of the Ukrainian Academy of Sciences. When it comes to completion of an extensive research project or job, the in-house computing capabilities of the institute often do not suffice. In such situations, the institute has access to the collective-user computing center at the exhibition.

92

Specialists are also studying here the best models of organization of the computational process; in this center they may more frequently exchange experience, listen to lectures, hold meetings and discussions. In parallel, the opportunity occurs for consolidating the practical base for increasing the State fund of algorithms and programs.

Other steps are necessary in addition. It still remains to perfect, with consideration of the specifics of computing centers, the cost-accounting indicators and the system of material incentives for all those who make able use of computers. The labor of programmers must be more quickly industrialized. With respect to cost, software is on a level with the computers themselves. Moreover, it is important to develop funds of algorithms and programs to make more extensive use of what has already been accumulated, and avoid duplication of efforts.

Of course, this process can be speeded up not only by finding new methods of using computers, but also by improving personnel training.

"The need for such personnel is very great" notes D. V. Golovko, secretary of the Kiev Party Commission. "Investigation of 230 enterprises and organizations that have had recourse to assistance from computer technology has shown that a considerable number of them have no service personnel of the necessary skill."

Clearly the inadequate pace of training engineers for utilizing computers ends up in miscalculations in the use of equipment. And the equipment will not tolerate a nonprofessional attitude.

A considerable effect is to be realized from further reduction of computer idle time, increased machine shift coefficient and other measures that are of assistance in the capable use of the inventory of computer equipment. Experience accumulated in the Ukraine in dispatcher information work, the use of standardized "machine-readable" forms of documents approved by the central statistical administration and finance agencies, and the organization of centralized technical servicing of the computers in the city are important advances on the path to putting this equipment into operation at full capacity.

6610
CSO: 1863/20

PROGRESS ON UKRAINIAN AUTOMATED PLANNING CALCULATION SYSTEM

Kiev RABOCHAYA GAZETA in Russian 23 Sep 82 p 3

[Excerpts from interview with V. N. Khalapsin, deputy chairman of UkSSR State
Planning Committee by RATAU correspondent in Kiev: "Computer Helps With Plan-
ning"; date of interview not given]

[Excerpts] The blue display screen energized in Kiev on 12 Kirov Street
has for the first time provided direct telecommunications between specialists
of Ukrainian State Planning Committee and computers at a computing center in
another part of the city. A new stage has opened up in setting up the republic
automated system of planning calculations (ASPR), enabling improvement of the
quality of plans and accelerated perfection of the fiscal mechanism. V. N.
Khalapsin, deputy chairman of the UkSSR State Planning Committee tells RATAU
correspondent about paths of solving this problem formulated by the Twenty-
Sixth CPSU Congress.

[Question] Apparently on the example of republic-wide automated management
systems (RASUs) we can get a simpler representation of the structure of the
statewide automated system for collection and processing of information for
planning and control (OGAS) as a whole and of the ASPR in the makeup of this
system?

[Answer] Of course, the republic-wide system with certain allowances can be
treated as a counterpart of the statewide system. Today the RASU includes
about eight hundred systems of three levels. The first comprises the automated
control systems (ASUs) of practically all enterprises and associations, many
of which include automated process control systems. The ASUs of the Ministry
of Ferrous Metallurgy, the Ministry of the Coal Industry, the Ministry of Light
Industry and of other ministries and agencies have become a second "story".
The third, upper, level is comprised of seven intersectoral systems. Most
important among these is ASPR of the UkSSR State Planning Committee.

This system is a kind of core of the RASU since all the major threads of man-
agement of the socialist economy of the republic converge in the State Planning
Committee. It is for this reason that the Party and the government are giving
such great attention to improving the quality of planning -- the heart of the
fiscal mechanism.

[Question] The resolutions of the Twenty-Sixth CPSU Congress formulate the task of not only introducing, but also effectively utilizing the automated system of planning calculations. What is the effectiveness of the ASPR today?

[Answer] Machine computations have become an organic part of our work, and are used on all stages and phases of planning development. Right now more than twenty percent of the forms and drafts of plans are put out totally in machine form and transmitted to the Main Computing Center of USSR State Planning Committee on computer media. Computers are used in solving more than 1,500 planning and economic problems, which would require ten times the staff for manual calculation.

No less important is the fact that the use of computer technology improves the quality of plans. It compresses the time of analysis, enabling comparison of several versions for an entire sector and selection of the best of these.

[Question] What is the outlook for development of the ASPR?

[Answer] In the present five-year plan, we intend to complete development of the first phase of the system, giving it properties of integrity and totality. This will enable a considerable increase in efficiency of the ASPR, since it gives the opportunity of much wider utilization of the logical-calculation capabilities of modern computers, applying them to extensive complex calculations, solution of the most important economic and social problems.

We will be able to utilize the capabilities of the ASPR to full extent with the maximum effect for the national economy only with establishment of regular intercomputer interaction with ASUs of all ministries and agencies within the framework of the republic-wide network of computer centers with data transmission system. And completion of the creation and perfection of all systems depends in large measure on the initiative and purposefulness of the respective sector-wide subdivisions.

6610
CSO: 1863/20

AUTOMATION OF DATA PROCESSING IN STROYBANK SSSR

[Excerpts from article by Yu. A. Chumachenko, chief, Main Computing Center, Stroybank SSSR]

[Excerpts] "ASU-Stroybank" is analogous to the hierarchical structure of the Stroybank SSR system, and is connected via automated control systems [ASUs] that are part of the Statewide Automated System for Collection and Processing of Information for Accounting, Planning and Control in the National Economy [OGAS] to the corresponding links of management of the national economy: to higher links -- via the automated systems of Gosplan SSSR, Gosstroy SSSR, Gossnab SSSR, Gosstandart SSSR, the USSR Ministry of Finances, Gosbank SSSR and so on; to intermediate links -- via the automated control systems of sectoral ministries and agencies and Soviet-wide industrial associations; to primary links -- via automated control systems in construction and industry. In addition, provisions have been made for linkage with the ASUs of territorial agencies of management of the national economy.

The hardware of "ASU-Stroybank" is based on YeS EVM computers. The software includes a set of unified programming languages and the OS YeS operating system. The database is designed around the use of unified classifier and vocabulary structures, as well as standardized forms of documents for ASUs of the different management levels. In our opinion, this is one of the most important directions of automation of accounting and bookkeeping in the Stroybank SSSR system.

The role of computers in processing banking information increases from year to year. In 1970, computers were used for data processing in six institutions of Stroybank SSSR. In 1981, the number had increased to 595. The use of computers in the computing centers of the USSR Central Statistical Administration, which have at their disposal a large number of third-generation computers, has become an important factor in the activity of institutions of the bank.

The interagency commission has accepted for industrial use programs that have been created by Moscow metropolitan, Ukrainian republic and Voronezh oblast offices of Stroybank SSSR. This has considerably expanded the number of bank institutions that use computers to process documents of the operational day. During the Tenth Five-Year Plan, the number of such institutions rose by a factor of more than 2.5, and made up 50% of the total number of departments

of the bank doing computational operations. By the beginning of the current five-year plan, computing centers were processing 75% of the documents of the operational day for institutions of Stroybank SSSR. Due to automation of the processing of operational accounting and bookkeeping information, we have a slowdown in the growth of the bookkeeping staff, 400 persons are laid off each year, the economic effect from introducing standard design features in the system has reached 3.6 million rubles.

Most extensively used in the Stroybank SSSR system is a program for processing documents of the operational day for the M-5010 keypunch-computer complex. This program was developed by the computing center in cooperation with specialists in bookkeeping at Voronezh Oblast Office. It has found application in 212 departments of the bank. Also being used is a program of the computing center of Gomel Statistical Administration for the Minsk-32 computer that has recently been replaced by programs of computing centers of the Moscow metropolitan and Ukrainian republic offices of the bank as developed for YeS-1022 third-generation computers of the unified series. In future, programs of these computing centers will completely replace some less efficient programs for second-generation computers with low degree of automation of bookkeeping operations.

Programs of the computing centers of Moscow metropolitan and Ukrainian republic offices of the bank are also being widely introduced because they permit the most complete automation of bookkeeping operations when doing calculations in construction with consideration of the capability of using a variety of up-to-date computers of the "Ryad" type.

The program complex of the computing center of Moscow Municipal Office of Stroybank SSSR is oriented toward large volumes of information -- up to 25,000 documents a day -- and is realized on YeS-1022 computers.

The program complex of the republic-wide computing center of Ukrainian Republic Office is intended for processing bookkeeping information on YeS-1022 computers using a disk operating system. In the immediate future, this complex will be improved with regard to use of an improved operating system and will be introduced in bank institutions with information volume of no more than 10,000 documents per day.

The program complex of the computing center of Voronezh Oblast Office is oriented toward small volumes of bookkeeping information of less than 3,500 documents per day, and is realized on the M-5010 keypunch-computer complex.

CONFERENCES

SEMINAR HELD ON INTERACTIVE SYSTEMS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 5, Sep-Oct 82 p 127

[Article by Leonina Nikolayevna Nekrasova, engineer; Ol'ga Leonidovna
Perevozchikova, candidate of physicomathematical sciences; both from the Institute
of Cybernetics, UkSSR Academy of Sciences (Kiev); and Mikhail Georgiyevich
Khoshtaria, candidate of physicomathematical sciences, Computer and Information
Center, GSSR Ministry of Finance (Tbilisi)]

[Text] The 4th school-seminar, "Interactive Systems," was held in Sukhumi from
10 through 27 May 1982. It was organized by the Council of Young Scientists of
the TsL [Central Laboratory] of the VLKSM [All-Union Lenin Komsomol] of Georgia,
the Council on Automation of Scientific Research under the Presidium of the USSR
Academy of Sciences, the Institute of Cybernetics of the GSSR Academy of Sciences,
the Computer Center of the GSSR Academy of Sciences, the Georgian NIINTI [Scien-
tific Research Institute of Scientific and Technical Information, Economic and
Technical Research], the Computer and Information center of the GSSR Ministry of
Finance, and the Institute of Cybernetics imeni V. M. Glushkov, UkSSR Academy of
Sciences.

Taking part in the work of the school were 262 scientists from 27 USSR cities.
Represented were more than 69 scientific centers and organizations of the USSR
Academy of Sciences and union republic academies of sciences, ministries and
departments (including 17 VUZ's from 9 union republics).

V. V. Chavchanidze, academician of the GSSR Academy of Sciences, delivered the
opening address. There were 27 papers read in 2 plenary sessions and 4 lecture
sessions in the school.

The papers read in the plenary sessions were: "Principles of Organization of an
Interactive System," by V. A. Lovitskiy (Kharkov); "Principles of the Theory and
Methods of Elaboration of Learning, Service and Interactive Systems," by A. M.
Dovgyallo (Kiev); and "Using Dialog in Synthesizers of Programs," by M. G.
Pkhovelishvili (Tbilisi).

The work of the school was conducted in three sections:
--principles of organization of dialog: models, systems and hardware;
--problem-oriented interactive systems; and
--information and logic support and intelligent data banks.

98

The discussion, "Linguistic Problems of Dialog in Natural Language," was held on the concluding day of the work of the school.

Considered in the school-seminar was a range of problems concerning the development, implementation and application of interactive systems, for example: "Experience of Development and Application of Automated Systems for Analysis of Situations and Prediction" (I. K. Tsikunov), "System for Interactive Debugging of Programs in a High-Level Language" (N. V. Shkut, G. S. Shinkevich, L. B. Kurakina and S. V. Zubritskaya), "FORTRAN Programming in the Interactive Mode with the Unified System of Computers" (Z. S. Brich and V. I. Tsagel'skiy), "The DINTES Time Sharing System" (P. B. Varapay, R. V. Gorbunova, E. V. Kovalevich and P. T. Chuprigina), "On User Interaction with Data Bases in Transportation Management Systems" (V. I. Gritsenko and V. A. Bogemskiy).

Much attention was paid to methodological issues in designing problem-oriented interactive systems, for example, the papers: "Organization of Computations in Problem-Oriented Systems" (Ye. L. Yushchenko, O. L. Perevozchikova and I. V. Krishtopa), "Some Trends in Evolution of Computer Capabilities" (V. D. Zhil'chenkov and A. P. Sokolov), "An Approach to Describing Interactive Systems" (V. M. Mikhelev and A. K. Petrenko), "Principles of Designing Multiuser Interactive Systems for Making Plan Decisions" (A. P. Pavlov and A. G. Vol'vakh).

Special interest was shown in discussing issues of software, linguistic and information support in these papers: "Software for the System of Small Computers" (V. P. Semik), "Some Issues in Dialog Interaction of Problem-Oriented System with an Automated System for Situational Analysis" (L. N. Nekrasova and V. S. Yakovleva), "Some Features of the Interactive Mode in Solving Symbolic Problems" (V. P. Klimenko, S. B. Pogrebinskiy and Yu. S. Fishman), "Implementation of an Interactive Procedure for Synthesis and Analysis of a Complex System" (Ya. K. Tenteris and V. Ya. Eulis), "An Approach to Developing a Precise Language of Linguistic Semantics" (A. V. Gladkoy and A. Ya. Dikovskiy), "Understanding of Text and Semantic Compression" (E. F. Skorokhod'ko), "Simulation of Natural Man-Computer Dialog (Application of Inflection Analysis of the 'Glokaya Kuzdra' Type)" (A. V. Anisimov and A. P. Beletskaya).

In the opinion of the participants, the school was rather impressive, the papers substantive and interesting, and read at a high scientific level. It should be noted that in selecting papers, preference was given to developments proven in practice on existing computers.

One can state that at the contemporary stage in the development of computer technology, the problem of man-machine communication and dialog is especially relevant. Modern interactive systems allow man to use the customary professional language (restricted natural or close to natural), which is considerably extending the sphere of computer applications, classes of problems being solved and range of users.

The school-seminar participants noted with satisfaction the specialization of many scientific centers in a particular field of design of interactive systems and the contemporary level of interactive systems being implemented.

The recommendation was made to hold the 5th All-Union School-Seminar in 1983.

HISTORY OF UKRAINIAN INSTITUTE OF CYBERNETICS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 5, Sep-Oct 82 pp 5-8

[Article: "Milestones in the History of the Institute (Dates, Figures and Facts)"]

[Text] 1947-1956

--The first domestic electronic machine, the MESM [small electronic calculator] is developed at the Institute of Electrical Engineering, UkSSR Aacademy of Sciences (placed into operation on 25 December 1951).

--The computer engineering laboratory (headed by V. M. Glushkov) is transferred to the Institute of Mathematics, UkSSR Academy of Sciences.

1957

--The Computer Center of the UkSSR Academy of Sciences is established with the status of a scientific research institute. V. M. Glushkov is named director.

--Foundations of the theory of digital automata are established.

--The address language, the first language for automatic high-level programming is developed.

--First series of lectures on mathematical programming and theory of computers are organized for engineers and computer design mathematicians.

1958

--Basic long-term programs for research in theoretical and applied cybernetics are defines.

--Coordinating commission on computational mathematics and computer engineering organized in UkSSR Academy of Sciences (chaired by V. M. Glushkov).

--First scientific conference on "New Developments in Computational Mathematics and Computer Engineering" held.

1959

--Basic research performed in theory of abstract automata, theory of computers and methods of automating programming. Concepts formulated for developing machines with a high level of input and internal lnaguages.

--The Kiev ETsVM [electronic digital computer] is placed in trial operation.

--Computer Center scientists take part for the first time in the work of IFAC [International Federation of Automatic Control].

--First programs are approved for systematic courses of lectures on cybernetic specialties for KGU [Kiev State University].

1960

--Basic results of analysis and synthesis of digital automata, which define the path of computer development from the process of design to industrial technology, are published.

--Efficient methods are developed for solving optimization problems for control, planning and design (successive analysis of alternatives, etc.) that are of major practical value (design of roads, efficient schemes for hauling freight).

--For the first time in Europe, an experiment is performed on the remote control of the process of melting steel (at the Metallurgical Plant imeni F. E. Dzerzhinskiy in Dneprodzerzhinsk).

--Philosophical analysis is made of the subject and methods of cybernetics which marked the beginning of its new interpretation (V. M. Glushkov). Subsequently, this interpretation was adopted as the basis in articles on cybernetics in the Ukrainian Soviet (1961), Great Soviet (1973) and British encyclopedias.

--Coordination meeting of union republic computer centers held on problem of "Computational Mathematics and Computer Technology."

1961

--The country's first general-purpose control computer, the Dnepr, which defined new paths in automating production and industrial processes, is developed and put into series production.

--Scientific Council on the Problem of "Cybernetics" is organized in the UkSSR Academy of Sciences.

--V. M. Glushkov's monograph, "Teoriya algoritmov" [Theory of Algorithms], is published.

1962

--UkSSR Academy of Sciences Computer Center becomes the UkSSR Academy of Sciences Institute of Cybernetics.

--The country's first small computer built with semiconductors, the Promin', is developed and placed in series production.

--Efforts to develop automated enterprise management systems are begun.

--Coordination is effected for efforts in the field of automata and self-adjusting systems, theory of mathematical machines and economic cybernetics within the framework of the academies of sciences of the countries in the socialist community.

--Institute scientists take part for the first time in the work of the IFIP-62 International Congress (Munich).

102

1963

--For the first time in the world, the concept of developing computer networks and hierarchical systems for managing the national economy, the OGAS [Statewide automated System], is advanced. Draft design is worked up for the state network of computer centers.

--Prototypes of industrial robots are manufactured (Avangard system, automaton for control of resistance welding, Strum machine).

--SKB [special design bureau] is organized with an experimental plant.

--The interdepartmental Scientific Council on the Introduction of Computer Technology and Economic-Mathematical Methods into the USSR National Economy began operating under the USSR Council of Ministers State Committee for Coordination of Scientific Research (chaired by V. M. Glushkov).

--Courses for mass study of computers and programming organized (in accordance with a 300-hour program).

--An All-Union Symposium on New Trends in Development of Computer Technology (in Uzhgorod) and the International Conference of the Countries in the Socialist Community on Algorithmic Languages and Automatic Programming (in Kiev) were held.

1964

--Lenin Prize awarded to V. M. Glushkov for series of works on theory of digital automata.

--For the first time in world practice, a small system is developed for computer-aided design of assemblies and units for digital computer equipment.

--V. M. Glushkov's monograph, "Vvedeniye v kibernetiku" [Introduction to Cybernetics] published.

--A complex of major technical and economic computations is effected for optimum management of construction of the Burshtynskaya GRES, a number of facilities for a chemical combine and the subway bridge across the Dnepr River, and for shipments of mass freight, which afforded a significant economic effect.

1965

--Apparatus of microprogram algebras developed.

--Methods of stochastic programming and automat modeling of complex systems developed.

--MIR-1 computer accepted by state commission and placed in series production. New ideas involving computer architecture and organization of computational process implemented. Domestic industry for small machines begins.

--Detailed design elaborated for the large computer, the Ukraina, which anticipated many ideas of the American large computers of the seventies.

--System developed for acquisition and expeditious processing of results of oceanological observations (for the Lomonosov, a scientific research vessel).

--First issue of KIBERNETIKA, the all-union scientific and theoretical journal, published (reprinted in the United States).

--V. M. Glushkov chosen as member of Program Committee for IFIP-65 (New York).

1966

—Principles of theory and hardware developed for distributed system for automatic control of processes in continuous media.

—Library of main programs for MIR computer established.

—Country's first republic bank of algorithms and programs established.

—School   of Cybernetics organized at KGU [Kiev State University] imeni T. G. Shevchenko. Prominent Institute scholars enlisted to lecture in courses on this subject.

1967

—Prizes awarded:
  imeni N. M. Krylov (UkSSR Academy of Sciences), for series of work on theoretical cybernetics (V. M. Glushkov);
  imeni Lenin Komsomol (All-Union Lenin Komsomol Central Committee), for development and introduction of automated control systems in industry (V. V. Shkurba and V. K. Kuznetsov);
  imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee), for series of work on digital computer software (A. A. Stogniy).

—Country's first automated enterprise management system for mass production, the L'vov, placed into operation.

—Analitik algorithmic language developed for analytic and literal transformations.

—Complex of computations performed for design of Druzhba petroleum pipeline.

—Department of Theoretical Cybernetics and Systems Analysis of the Moscow Physico-Technical Institute established at the Institute.

—Construction of UkSSR Academy of Sciences Cybernetic Center begun in Teremki. All-Union Lenin Komsomol Central Committee declares it an All-Union Komsomol key construction project.

—First domestic computer for automation of control of electron-ion processes, the Kiev-67, developed and handed over to an interdepartmental commission.

1968

—Prizes awarded:
  USSR State, for development of principles of design of structures of small computers for engineering computations and software incorporated in the MIR computer (V. M. Glushkov, S. B. Pogrebinskiy, V. D. Losev, A. A. Letichevskiy, Yu. V. Blagoveshchenskiy, I. N. Molchanov and A. A. Stogniy);
  imeni the Lenin Komsomol (All-Union Lenin Komsomol Central Committee), for series of work on theory of automata, practical methods of synthesis and computer design automation (Yu. V. Kapitonova and A. A. Letichevskiy):
  imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee), for development and introduction in series production of the Iskra EKVM [electronic keyboard computers] (G. I. Korniyenko—in a group of authors).

—Methods elaborated for expert assessments in scientific and technical forecasting in sectors of the national economy.

--First All-Union Conference on "Computer Programming Automation Systems" held.

--V. M. Glushkov, member of the IFIP-68 (in Edinburgh) Program Committee, heads the section on "Computer Applications in the Natural Sciences, Engineering, Linguistics and Library Sciences. Artificial Intelligence."

1969

--Institute awarded the Order of Lenin for outstanding progress in development of science and training of scientists.

--Academician V. M. Glushkov awarded the title of Hero of Socialist Labor.

--MIR-2 computer accepted by the State Commission and put into series production.

1970

--UkSSR State Prize awarded for development and introduction of the L'vov automated system (V. M. Glushkov, V. I. Skurikhin, V. V. Shkurba, V. K. Kuznetsov, T. P. Podchasova and A. A. Morozov--in a group of authors).

--USSR's first multimachine system with a peripheral computer center, the Abonent, which marked the beginning of establishing multiuser computer centers, is developed.

--The Ros' desktop electronic keyboard computer developed for automation of high-volume computations.

--V. M. Glushkov chosen as foreign member of Leopoldina Academy (GDR).

1971

--Prizes awarded:
  imeni N. M. Krylov (UkSSR Academy of Sciences), for series of work on research in methods of optimization (V. S. Mikhalevich);
  imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee), for series of work on numerical methods in solving problems of optimal control (B. N. Pshenichnyy).

--Efforts begun to implement in practice the ideas and designs for establishing the OGAS [Statewide Automated System], GSVTs [State Network of Computer Centers], and RAS [Republic Automated System for Collection and Processing of Information for Accounting, Planning and Management in the Economy] of the Ukrainian SSR.

--MIR Computer Users Association and first sectorial problem laboratory organized.

--V. M. Glushkov becomes consultant to government of People's Republic of Bulgaria on problems of establishing automated control systems and comprehensive automation of production.

--V. M. Glushkov appointed chairman of IFIP-71 (in Lublin) Program Committee.

1972

--UkSSR State Prize awarded for development and introduction of an automated system to control electroplating lines in a protective coating shop in an enterprise with a multi-nomenclature nature of production (A. I. Nikitin, I. V. Sergiyenko and A. I. Stiranka--in a group of authors).

--Institute becomes coordinator of research on automation of large-scale integrated circuit design.

--The Kiev-70, a specialized control computer with integrated elements, which allowed assimilation of highly efficient industrial processes in microelectronics, is developed.

--V. M. Glushkov's monograph, "Vvedeniye v ASU" [Introduction to Automated Control Systems], published.

--Soviet-American seminar held on problems of designing organization structures in management.

1973

--Prizes awarded:
  UkSSR State (A. A. Bakayev, Yu. M. Yermol'yev, T. P. Mar'yanovich, V. S. Mikhalevich, N. Z. Shor and N. V. Yarovitskiy);
  imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee), for development and introduction of problem-oriented complexes of automated control systems in the national economy (A. A. Morozov, A. A. Kobozev and V. I. V'yun).

--V. M. Glushkov becomes chairman of the Scientific Council on Computer Technology and Control Systems under the State Committee of the USSR Council of Ministers for Science and Technology and the Presidium of the USSR Academy of Sciences.

--The thousandth patent application for an invention is recorded.

--World's first "Entsiklopediya kibernetiki" [Encyclopedia of Cybernetics] is published in two volumes.

--Computer-aided system for design of computers and software (PROYEKT) is developed and placed in trial operation.

--Standard automated system for controlling an enterprise with a multi-nomenclature nature of production (Kuntsevo system) is developed.

1974

--Specialized computer system for checking and forecasting quality of resistance welding, the Svarog, is developed and introduced.

--New nonhomogeneous multimachine complex of the MIR-2 and BESM-6 is placed into operation.

--Academician V. M. Glushkov becomes foreign member of Bulgarian Academy of Sciences.

--At the IFIP-74 Congress (in Stockholm), V. M. Glushkov presented the concept of designing a "non-Neumann type" of computer.

1975

--As the victor in All-Union Socialist Competition, the Institute is awarded the travelling Red Banner and Diploma of the CPSU Central Committee, the USSR Council of Ministers, the All-Union Central Council of Trade Unions and the All-Union Lenin Komsomol Central Committee.

--Models of tunnelling cryotrons are manufactured for the first time in domestic practice.

--Country's first general-purpose microcomputer developed at the Svetlana NPO [Scientific Production Association] (in Leningrad) with Institute participation.

--General-purpose system for modeling robots with elements of artificial intelligence developed.

--Ritm-3, automated system for immediate display of situation, placed in industrial operation.

--V. M. Glushkov becomes foreign member of GDR Academy of Sciences, honorary foreign member of Polish Cybernetics Society, and honorary doctor of Dresden Engineering University.

1976

--Institute is entered on Republic Board of Honor for achieving highest indicators in socialist competition and successful completion of national economic plan for 1976.

--UkSSR State Prize awarded for development of system for automatic checking of quality of parts in aviation gas-turbine engines, the Tsikl system (Yu. T. Mitulinskiy, G. I. Korniyenko, S. K. Lesnichiy and A. S. Odinokiy--in group of authors).

--The Displan experimental automated system for balance computations with optimization of limited resources accepted by the Interdepartmental Commission.

--First phase of automated system of plan calculations (ASPR) for UkSSR Gosplan developed jointly with UkSSR Gosplan GlavNIIVTs [Main Scientific Research Institute Computer Center] is accepted by the State Commission.

--Automated system for control of industrial process of glass-tube production (in Saransk) accepted by Interdepartmental Commission and introduced.

--The Mioton device which makes it possible to rehabilitate motor function disorders by using program control of human muscular activity is put into series production.

--The Ekspan base specialized computer complex with fourth-generation elements is developed.

--Programmer's process system to produce programs (RTK) developed.

--V. M. Glushkov's monograph, "Makroekonomicheskiye modeli i printsipy postroyeniya OGAS" [Macroeconomic Models and Principles for Designing the Statewide Automated System], published.

--The TEMP-EK standard system for processing results of flight tests is accepted by the State Commission.

1977

--Prizes awarded:
  USSR State, for series of work on the theory of discrete converters and methods of automating the design of electronic computers, which have been applied in active systems (V. M. Glushkov, V. P. Derkach and Yu. V. Kapitonova);
  USSR State to A. A. Morozov (in a group of authors);
  UkSSR State, for development and introduction of efficient methods and means of automated control of ferrous-metallurgy enterprises in UkSSR (F. I. Andon and V. I. Gritsenko--in a group of authors);
  UkSSR State to B. N. Malinovskiy (in a group of authors);
  imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee) for development and implementation of software for computer-aided design systems (V. I. Dianov, M. I. Dianov, V. M. Mikhaylov, N. P. Malevanyy and V. V. Gayduk).

107

--For the first time in our country, (within the scope of the effort on developing the GSVTs [State Network of Computer Centers]) information is exchanged between remote computer centers (UkSSR Academy of Sciences Institute of Cybernetics in Kiev and the VNIIPOU [All-Union Scientific Research Institute of Problems of Organization and Control] in Moscow) by using series equipment.

--State Commission accepts country's first problem-oriented engineering system, the Mars ASUP [automated production control system].

--State Commission accepts the Chertezh computer-aided design system for specialized complex objects.

--The Etalon problem-oriented computer is sent for series production after completion of development.

--First domestic monograph on computer networks, "Seti EVM" [Computer Networks], edited by V. M. Glushkov, is published.

--V. M. Glushkov becomes foreign member of Polish Academy of Sciences.

--Standard high-speed system for data reception and transmission in automated control systems for various purposes is developed. First licensing agreement is signed.

1978

--For achieving the highest results in the All-Union Socialist Competition, the Institute is declared the winner and is awarded the travelling Red Banner of the CPSU Central Committee, the USSR Council of Ministers, the All-Union Central Council of Trade Unions and the All-Union Lenin Komsomol Central Committee and entered on the All-Union Board of Honor at the Exhibition of Achievements of the USSR National Economy.

--Academician V. M. Glushkov is awarded honorary title of Honored Scientist of the Ukrainian SSR for outstanding service in development of domestic science and training scientists and for active social activity.

--Prizes awarded:
  UkSSR State, for the "Encyclopedia of Cybernetics" in two volumes;
  UkSSR State, for development and introduction in the national economy of the highly efficient problem-oriented system for automating tests of complex prototypes of new technology (P. M. Siverskiy, V. I. Zabolotnyy, O. V. Marchenko and Yu. N. Rukhlyadev);
  imeni S. A. Lebedev (UkSSR Academy of Sciences), for series of work on development of structures and principles of design of general-purpose and control computers (Z. L. Rabinovich and B. N. Malinovskiy).

--New class of integrated circuits, LSI diode-matrix circuits which exceed world achievements in their parameters, is developed and accepted by the State Commission.

--Country's first program-computer complex, the PIRS, of a multichannel automated system for processing experimental data from full-scale and field tests of new equipment is developed.

--Practice of the Institute's work in selecting and training scientific personnel is approved by the Presidium of the Higher certification Commission under the USSR Council of Ministers.

--BARS system is awarded a gold medal at the Leipzig Fair.

1979

--Institute is named the winner in the All-Union Socialist Competition and awarded the travelling Red Banner of the CPSU Central Committee, USSR Council of Ministers, the All-Union Central Council of Trade Unions and the All-Union Lenin Komsomol Central Committee and entered on the All-Union Board of Honor at the Exhibition of Achievements of the USSR National Economy.

--Prizes awarded:
   imeni S. A. Lebedev (UkSSR Academy of Sciences) to Academician V. M. Glushkov for work on the theory of advanced computers and development of high-throughput hardware for computer equipment and control systems;
   USSR State, to Academician of UkSSR Academy of Sciences I. N. Kovalenko (in a group of authors);
   UkSSR State, for development and introduction of new technology in design and control based on third-generation computers (Ya. G. Verenko, V. I. V'yun and A. A. Kobozev--in a group of authors);
   UkSSR State, to I. V. Vel'bitskiy (in a group of authors);
   imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee), for series of work on elaboration and theoretical basis of algorithms for solving problems of artificial intelligence with computers (V. V. Bublik, A. B. Godlevskiy, A. P. Zhezherun, V. F. Lyabakh and V. F. Chuykevich).

--Automated system developed for recognizing words and continuous speech, adjustable to user's voice, with an advanced vocabulary (up to 1,000 words).

--Basic automated system developed for controlling processes of transportation and storage of petroleum in main oil pipelines with any configuration.

--The TEST hybrid system for control of nonstationary vibration tests is developed.

1980

--Institute wins All-Union Socialist Competition and is awarded travelling Red Banner of the CPSU Central Committee, USSR Council of Ministers, the All-Union Central Council of Trade Unions and the All-Union Lenin Komsomol Central Committee; for results of 10th Five-Year Plan, Institute is awarded Memorial Badge of the CPSU Central Committee, USSR Council of Ministers, All-Union Central Council of Trade Unions and the All-Union Lenin Komsomol Central Committee "For High Efficiency and Quality of Work in the 10th Five-Year Plan" and entered on the All-Union Board of Honor at the Exhibition of Achievements of the USSR National Economy.

--Prizes awarded:
   imeni A. N. Krylov (USSR Academy of Sciences), to Academician V. M. Glushkov for series of work on methods of optimization in planning and management;
   UkSSR State, for development and introduction in the national economy of basic process control system for main oil pipelines (D. V. Karachenets, Ye. P. Pozdnyakov and A. M. Ovcharenko (in group of authors);
   imeni S. A. Lebedev (UkSSR Academy of Sciences), to A. I. Kondalev (in group of authors) for series of work on methods and means of analog and hybrid computations;
   imeni Lenin Komsomol (All-Union Lenin Komsomol Central Committee), for development and introduction of facilities of cybernetic technology for data processing and process control in industry (G. B. Vavilin, A. I. Zayonkovskiy, B. V. Novikov, V. A. Petrukhin and V. V. Syrov).

--First phase of the Ukrainian SSR Republic Automated System completed together with the UkSSR Gosplan GlavNIIVTs [Main Scientific Research Institute Computer Center] and head organizations of the sector ministries and departments and accepted by the State Commission.

--New methodology of systems optimization is proposed which results in efficient resolution of a broad class of economic and production process problems.

1981

--Institute wins All-Union Socialist Compettion of Groups of Scientific Institutions of the USSR Academy of Sciences and Union Republic Academies of Sciences and is awarded the travelling Red Banner of the USSR Academy of Sciences and Central Committee for the Education, Higher Schools and Scientific Institutions Workers Union.

--Prizes awarded:
 USSR State, for series of work on development and widespread introduction of modern mathematical methods of optimization, published in 1962-1979 (V. S. Mikhalevich, A. A. Bakayev, Yu. M. Yermol'yev, I. V. Sergiyenko, V. L. Volkovich, B. N. Pshenichnyy, V. V. Shkurba and N. Z. Shor);
 UkSSR State, to Academician V. M. Glushkov (in group of authors);
 USSR Council of Ministers, for development and introduction in the national economy of hardware and software for developing multilevel automated systems for data acquisition, transmission and processing (BARS system);
 USSR Council of Ministers, for development and introduction of a complex of standard, problem-oriented, interactive facilities for computer-aided design of assemblies in radioelectronic apparatus at automated work stations;
 imeni S. A. Lebedev (UkSSR Academy of Sciences), for development of computer storage elements and devices (G. A. Mikhaylov--in a group of authors);
 imeni N. Ostrovskiy (Ukrainian Lenin Komsomol Central Committee), for series of work, "Development and Introduction of Production Complex of Hardware and Software of Cybernetic Technology to Automate Processes and Scientific Experiments" (N. I. Alishev, Yu. A. Brayko, L. B. Malinovskiy and S. I. Tret'yakov).

--Automated process system is placed in operation; it implements progressive processes of electron-lithographic production of complex microelectronic structures for computers with enhanced technical and economic parameters.

--Automated system is developed for ensuring safe sea navigation under the conditions of a conflicting navigational situation (Antikon).

COPYRIGHT: Izdatel'stvo "Naukova dumka" "Upravlyayushchiye sistemy i mashiny", 1982

8545
CSO: 1863/37

PUBLICATIONS

TABLE OF CONTENTS FROM 'CONTROL SYSTEMS AND MACHINES', SEPTEMBER-OCTOBER 1982

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 5, Sep-Oct 82 pp 1-2

112

Experience in Development and Introduction of Automated Control Systems

8545
CSO: 1863/37

INFORMATION SECURITY AT COMPUTING CENTERS AND IN AUTOMATED CONTROL SYSTEMS
(FROM NON-SOVIET PRESS DATA)

[Article by V. A. Gerasimenko]

[Text] In recent years considerable attention has been given in the non-
Soviet press to problems of security of information being stored and processed
at computing centers and in automatic control systems. Information security
in this sense is understood to mean the use of special facilities and tech-
niques aimed at preventing distortion, erasure and unauthorized use of infor-
mation. The major concern among non-Soviet specialists is caused by problems
of guarding information from unauthorized use. This is because of the high
vulnerability of information in present-day automated control systems and
computing centers, i. e. the comparatively wide possibilities for covert ac-
cess to such information. The Soviet press has called attention to the serious-
ness of this problem in Ref. 1, 2.

CHARACTERISTICS OF VULNERABILITY OF INFORMATION IN MODERN AUTOMATED CONTROL
SYSTEMS AND COMPUTING CENTERS. The high vulnerability of information in
automated control systems and computing centers (now a generally recognized
fact) has been a consequence of the objective peculiarities of development
of facilities, methods and forms of automating data processing. Of greatest
significance are the following:

--abrupt expansion of the variety and inventory of hardware for automation
of data processing;

--extensive proliferation of systems for large-scale automation of data pro-
cessing and control with wide spatial dispersion of the objects of automation;

--a general conversion to shared forms of utilizing automation facilities and
methods (collective-user computing centers, computer networks);

--extensive proliferation of integrated forms and methods of automated data
processing (automated data banks);

--complication of modes of operation of automation hardware (multiprogram,
multiprocessor, real-time, time sharing).

114

All this has led to a situation where large volumes of information are being concentrated and stored for a prolonged period on computer media, data of various organizational origin and functional designation are being brought together in unified databases, and large numbers of users are gaining direct access to the resources of automated control systems and computing centers (including to the information being processed). Also of fundamental importance is the fact that intensive data transmission takes place in present-day systems (particularly with transition to computer networks) over automated communication lines, including transmission to great distances.

It is these circumstances that have led to an abrupt increase in information vulnerability, i. e. possibilities for unauthorized (including malicious) access to this information. This means that unless special steps are taken in modern automated control systems and at computing centers, there is potentially a real threat of persistent information leaks. It is the elimination of this threat that makes up the major task of information security in the most general formulation.

Information vulnerability in automated control systems and at computing centers shows up in the presence of a certain number of channels for leakage of this information, a leakage channel being defined as objectively existing capabilities for unauthorized and unsupervised retrieval of data in a system. The information leak may be accidental, i. e. a consequence of accidental mistakes by operators or breakdowns of equipment and programs, or may be the result of malicious actions. The latter case is of particular concern to specialists since non-Soviet experience has shown first of all that nearly all recorded cases of unauthorized access to information have been precisely of this type, and secondly, the consequences of such a leak are much more serious than the accidental case, while in the third place it is considerably more complicated and expensive to protect against malicious actions than against accidental leaks.

It is for this reason that information security in the non-Soviet press usually means security against malicious actions, with the assumption that the guarantee of such security will reliably prevent accidental leakage of information as well.

The entire set of potentially possible leakage channels can be divided into three groups:

--channels that allow unauthorized access to information from the side, without direct access to the resources of the automated control system or computing center. This refers to possibilities for examining or photographing (via windows or doors of enclosures) printouts or graphic displays; use of listening devices; registration of electromagnetic emissions generated by various devices of automated control systems or computing centers, and crossfire that arises in adjacent lines, and also collection of trash carried out of the automated control system or computing center;

--channels associated with direct access to rooms of the automated control system or computing center, but not involving a change in components of the

115

system. This group includes stealing or copying information media; observation of data during processing with the purpose of memorizing or recording in some way; picking up trash that contains information being processed; intentional readout of information in fields of the memory unit during processing; illegal use of the terminals of registered users; posing as registered users by appropriating or figuring out keywords and other requisites for controlling access to information, and also gaining access to information by using so-called loopholes, i. e. opportunities for getting around the mechanism of access control as a consequence of imperfection and ambiguities of programming languages and public components of software (operating systems, database control systems and the like);

--channels involving a change in the structure of components of the automated control system or computing center. These channels include illegal connection to devices of the system or to communication lines of special recording equipment; malicious alteration of existing programs in such a way that these programs will realize unauthorized collection and recording of priveleged information in addition to executing the principal processing functions; development and unauthorized input into the computer memory of programs that organize and implement unauthorized access to information; malicious crippling of the security mechanism.

The above-formulated problem of information security can be specifically defined as covering with required reliability all leakage channels that are potentially possible in secured automated control systems or computing centers.

FACILITIES FOR SECURING INFORMATION IN AUTOMATED CONTROL SYSTEMS AND AT COMPUTING CENTERS FROM UNAUTHORIZED ACCESS. As of now, quite an impressive arsenal of various means has been developed that specialists say can be used to organize and ensure fairly reliable security of information in present day automated control systems and computing centers.

The set of existing means for securing information is determined by the possible methods of preventing unauthorized access to information stored or being processed in automated control systems or at computing centers.

The main methods of security are obstruction, control, masking, regulation, compulsion and incentive.

Obstruction involves setting up a physical barrier to the intruder on the way to the protected information: preventing penetration into the area and rooms of the automated control system or computing center, access to equipment, information media and the like.

Control is a method of protecting information by regulating the use of all resources of the system (hardware, programs, database components) within the scope of a set rule. It is assumed that in the automated control system or computing center there are established precise and unambiguous work rules for users, personnel, hardware, programs, database components and information media.

For users, there should be specified days of the week and hours of the day during which work is permitted, terminals from which access is allowed, database

to which access is permitted, a list of procedures (jobs, programs and the like) that are allowed for use.

For the personnel of automated control systems and computing centers, there are specified days of the week and hours of the day during which work is permitted, a list of resources of the system to which access is allowed, the order of access to resources.

For hardware, there are specified days of the week and hours of the day during which work is permitted, a list of persons having rights of use.

For programs and functional jobs, there are specified days of the week and hours of the day during which use is allowed, a list of terminals and the names of users who have a right to operate them.

For database components there are specified days of the week and hours of the day when use is allowed, a list of users who have the right of access, and a list of procedures permitted for use on these components.

For data media there are strictly defined storage locations, a list of persons having rights of withdrawal, and a list of programs that have rights of access to media.

Access control proper is implemented by execution of the following security functions:

--identification of users, personnel and resources of the system, where identification is understood to mean assignment of a personal identifier (name, code, password, etc.) to each object and subject mentioned above, and recognition (authentication) of the subject or object according to the presented identifier;

verification of authority, consisting in checking accord of weekday, time, and also requested resources and procedures to the established rule;

--permission and setting up conditions for work within the limits (and only within the limits) of the established rule;

--registration (documentation) of accesses to restricted information;

--response (suspending operation, refusal, detouring, signal) upon attempts at unauthorized actions.

Masking is a protection method in which the data to be secured are transformed so that their content is accessible only upon presentation of certain specified information and implementation of inverse transformation. Such transformations are called encryption, and in recent years outside the USSR they have been quite widely proliferated for information security in automated control systems and computing centers

Specialists consider encryption quite effective both from the standpoint of security proper, and from the standpoint of clarity for users. This form of

117

security can be used both during processing of the information to be secured and during storage, and when information is to be transmitted over a considerable distance over communication lines, it is the only method for reliable protection of transmitted data.

Encipherment or coding of data is used for encryption [Ref. 3]. In encipherment, each symbol of the data is enciphered sequentially; in coding, sections of the text having semantic significance are replaced by numeral, alphabetic or alphanumeric codes. In addition, some methods of data compression have a certain force of encryption.

Encipherment of data is most extensively used outside the USSR, recommended methods being [Ref. 4] encipherment by substitution, permutation, analytical transformation and ranging.

Substitution consists in replacing symbols of the plaintext with symbols of another alphabet or other alphabets. Simple monoalphabetic, ordinary polyalphabetic and monophonic polyalphabetic substitution are distinguished. Permutation is accomplished by changing the positions of the symbols in the plaintext in accordance with certain rules. Permutation may be simple, compound and special.

Encipherment by analytical transformation consists in transforming the symbols of the plaintext in accordance with certain analytical relations (such as rules of matrix algebra). Finally, ranging is a method of encipherment in which a symbol of a special random sequence called a range is superimposed on each symbol of the plaintext (e. g. according to addition with respect to a given modulus).

A major characteristic of the protective properties of encryption is the resistance of the cipher, by which we mean the minimum volume of ciphertext in which statistical characteristics of the plaintext may show up. Simple forms of encipherment (simple substitution, simple permutation) have low resistance and can be used only for enciphering short messages. Monophonic substitution, special forms of permutations, and ranging (assuming a long range is used) have high resistance. In practice, combined methods of encipherment are often used, where the plaintext is sequentially enciphered by two or even three (most frequently different) methods of encipherment. The overall resistance in this case will be at least the product of the resistances of the ciphers used.

A major operational feature of the given method of encryption is the productivity of encipherment, the more resistant ciphers as a rule having lower productivity, although there is no direct relation between these characteristics.

Also bespeaking the proliferation of encipherment in automated control systems and computing centers outside the USSR is the fact that, at least in the United States for example, a national standard for encryption was adopted several years ago to be realized by series-produced special hardware that can be interfaced with any device of an automated control system or computing center.

More detailed information on the principles of data encryption can be obtained from Ref. 4, 5 and elsewhere.

Regulation as a security method consists in setting up sets of measures to be realized during operation of automated control systems and computing centers so as to bring about conditions of automated processing and storage of priveleged information in the automated control system or computing center whereby opportunites for unauthorized access are minimized.

Specialists tell us that effective security necessitates strict regulation of the structural layout of automated control systems and computing centers (architecture of buildings, equipment of rooms, placement of hardware and so on), flowcharts for automated processing of protected information, organization and supervision of work of all personnel involved in data processing and so on.

Compulsion is a security method in which users and personnel of automated control systems and computing centers are forced to observe rules of processing and utilizing restricted information under threat of material, administrative or criminal responsibility.

Incentive is a security method in which the users and personnel of automated control systems and computing centers are internally motivated (i. e. by moral, ethical, psychological or other factors) to observe all rules of information security.

The above considered methods of information protection in automated control systems and computing centers are realized by application of a variety of means of security differentiated in accordance with the classification accepted in the non-Soviet press as physical, hardware, software, organizational, legislative and moral-ethical means.

Let us briefly consider the essence of each of these groups.

Physical means of security are mechanical, mechanical-electronic, or electro-mechanical devices and structures especially designed to present physical obstacles on the way to the protected information. These facilities include locks on doors, bars on windows, checkpoint equipment, intruder alarm systems and so on.

Hardware means of security are various electronic devices included in the hardware of the automated control system or computing center that perform certain protective functions (independently or in conjunction with other facilities). At the present time, non-Soviet systems utilize a considerable number of hardware facilities in nearly all structural components of automated control systems and computing centers: terminals, group I/O devices, central processors, external storage units and peripheral equipment.

Most widely used in terminals are devices designed for preventing unauthorized activation of the terminal (locks, interlocking and so on) and for user identification (devices for reading a code from cards, acoustic signal receivers, devices for length measurement and even fingerprint recognition and some other devices).

Devices for group control of I/O data that handle data exchange between a group of terminals and a central computer are using terminal address registers,

119

registers of the limits of the memory unit assigned to the terminal or user, registers for control of the level of secrecy of communication channels, circuits for checking the code of the communication channel and some others.

The principal hardware means used in central processors are: schemes for bimodal operation and priveleged commands, registers of memory limits, page sharing of memories, special secrecy bits, circuits and registers for key protection of data, associative memories, circuit points of memory input, etc. The main hardware means of data security in devices for control of external storage units is provided by read-only circuits.

Finally, peripheral equipment utilizes a variety of monitoring and interlocking devices. An independent and quite widely used group of hardware facilities is facilities for encipherment of data circulating in the process of operation of automated control systems and computing centers. Besides, large automated control systems and computing centers use some auxiliary security hardware: devices for erasing information on magnetic media, devices for signaling unauthorized actions and so on. Most of the enumerated devices are discussed in Ref. 6.

Software security means are special programs included in the software of the automated control system or computing center that are capable of protection functions. In view of their versatility, comparative simplicity of realization and flexibility, software facilities are an important and irreplaceable part of the mechanism of protection of present-day automated control systems and computing centers.

Program protection is conventionally divided into groups in accordance with the functions that they perform:

—identification programs, i. e. programs of recognition (of a terminal or user);

—programs for controlling operation (of hardware, users, jobs, database components);

—programs for delimiting access (to jobs routines, programs, database components);

—data encryption programs;

—programs for protecting programs (operating systems, database control systems, user programs);

—auxiliary programs for destroying residual information, forming a signature of secrecy of output documents, logging, simulating work with an eavesdropper (to divert attention), test monitoring of the security mechanism and the like.

The content of these groups of programs is discussed in Ref. 7.

Organizational means of security are organizational-technical and organizational-legal steps implemented in the process of setting up and operating

an automated control system or computing center to ensure information security. Organizational steps cover all structural components of the automated control system and computing center on all stages of their life cycle: construction of rooms, design of the system, installation and adjustment of equipment, testing and verification, utilization.

On the stage of setting up an automated control system or computing center, the following measures are implemented: accounting for requirements in developing the plan of the system and its structural components, in building and equipping the rooms, in developing the mathematical, program, informational and linguistic software, in the installation and adjustment of equipment, testing and certification of the system. Special emphasis during this stage is placed on determination of the real capabilities of the security system, for which an entire series of tests and checks is recommended.

Among the principal groups of steps implemented on the stage of operation of the automated control system and computing center are organization of the enabling mode, organization and support of controllable technology for automated processing of secure information, organization of work in computing center shifts, distribution of requisites for delimitation of access (passwords, profiles of authentication and the like), organization of logging and analysis of work protocols and the like.

In addition, a group of measures of general nature is distinguished: consideration of security in selecting and training personnel, organization of planned and surprise checks of the security mechanism, security planning and so on.

The content of the enumerated steps is discussed in Ref. 8.

Among the legislative means of security are legislative acts of the nation that set rules for using and processing information with limited access and establish measures of responsibility for violation of these rules.

Among the moral-ethical means of security are all kinds of norms that have grown up by tradition or that arise as computers are proliferated in a given nation or society. These norms for the most part are not obligatory like legislative steps; however, failure to conform generally leads to loss of authority and prestige of a person or group of persons (organizations).

Moral-ethical standards may be either unwritten (such as generally accepted standards of honor, patriotism and the like), or written, i. e. formulated in some code of rules and prescriptions. The most typical example of written standards is the code of professional behavior of members of the Association of Computer Users of the United States [Ref. 2].

All these means of security are divided into formal and informal means. Formal means are those that perform their protective functions strictly formally, i. e. according to a procedure provided beforehand and without direct participation of man. Informal means are those that are either determined by goal-directed activity of people or control such activity (directly or indirectly).

121

The necessity of using both formal and informal means is dictated by the fact that neither of these classes ensures reliable security by itself. The reason for this is the multifactor nature and nonformalizability of the security problem itself. As to the relation and significance of the distinguished classes of means, non-Soviet specialists are of one accord in saying that informal means play the decisive role. Some specialists state that in the overall security mechanism, 60-80% of the share falls to informal mechanisms, although the method of arriving at this quantitative measure is difficult to define.

PRINCIPLES AND METHODS OF ORGANIZING INFORMATION SECURITY IN AUTOMATED CONTROL SYSTEMS AND COMPUTING CENTERS. The security problem initially reduced to reliable delimitation of the access to protected data in automated control systems or at computing centers. This job was handled in the following way [Ref. 9].

Upon reception of a request for processing or printing out protected data, the first step was authentication (identification) of the client making the request. To do this, the latter, in submitting the request (or upon special demand of the program of the security mechanism) introduced identifying information (most frequently a codeword). The security programs compared this information with a reference stored in the computer memory. Upon a positive result of identification, the authority of the client was verified, i. e. the rights of access to the data specified in the request. Only upon positive result of the authority verification was the request granted. If any one of the checks gave a negative result, the request was denied, and the violator might face penalties: a fine in the form of a time delay, removal of the assignment from processing, disconnection of the terminal, informing supervisory agencies, transmission of an alarm signal and the like.

In some cases, in addition to these mechanisms, provisions were made for destroying so-called residual information, i. e. information remaining in the fields of the memory unit and registers after filling requests, and registration of accesses to protected information, and when handling information with high secrecy signatures, encryption was provided (mainly by enciphering data).

All these procedures may be realized by purely formal means (chiefly by software within the framework of the operating system and database control system), and therefore the security mechanism is also strictly a formal component of the automated control system or computing center. Such a mechanism, including all or part of the given procedures, is provided in most operating systems and database control systems.

However, practice has shown the inadequacy of these means; their presence is absolutely no guarantee against information leaks, which is what explains the considerable information theft reported in the non-Soviet press.

There have been attempts to reinforce the security mechanism within the framework of the operating system and database control system. The most typical example of such attempts is an operating system with expanded security functions known as a reserve safety system. The following security functions are realized in the system:

--identification of all users and structural components of the automated control system or computing center;

--control of access to data being stored and processed by authentication profiles; capabilities for handling authentication matrices of large dimensionality are provided;

--destruction of residual information in all fields of the memory unit that are used in the process of filling user requests;

--monitoring the processing of protected data by a specially allocated terminal;

--signaling in the event of attempts at unauthorized actions;

--keeping logs of accesses to protected data (so-called registration journals);

--preparation of reports based on the data of registration journals.

In addition to software means, a whole set of organizational measures has been suggested, and designation of a special responsible party for general organization supervision and coordination of performance of security functions.

However, during detailed tests of the system, a number of design flaws were discovered (e. g. the possibility of allocating control blocks in memory fields accessible to users) and some deficiencies were found, analysis of which led to the conclusion that it is inadvisable to concentrate all security means within the framework of operating systems and database control systems, i. e. within the scope of public software components. It was found that fundamental approaches and methods of organizing information security are decisively influenced (in addition to the peculiarities of present-day automated control systems and computing centers) by the fact that unauthorized use of information is principally a consequence of malicious actions. and the most typical feature of actions on the part of people is informality and unpredictability. As a consequence it has not been possible (and it has even been suggested that it is impossible in principle) to develop a strictly formal model of possible unauthorized actions that on the one hand would be sufficiently general, and on the other would be equivalent to a potentially possible real process.

On the basis of available experience, the following fundamental principles have been stated for organizing information security in automated control systems and computing centers [Ref. 10]: systems approach, specialized nature and informality.

The main requirements of the principle of a *systems approach* boil down to the fact that in order to ensure reliable information security in present-day automated control systems and computing centers, reliable and coordinated security must be provided on all their structural components, on all technological segments of automated data processing and throughout the time of operation of the automated control system or computing center. In accordance with this, the security mechanism must handle four macroroutines: protection of the

123

system from users; protection of users from one another; protection of users from themselves (i. e. from the users' own errors); and protection of the system from itself (i. e. from internal errors and failures of systems components).

The essence of the first macroroutine reduces to the fact that the security mechanism must be such that while providing the user with all capabilities for automated data processing, possibilities for unauthorized action on the part of the user are prevented: access to data of other users, access to system-wide data, access to supervisory areas, possibility of changing general-use programs (especially system-wide components), possibility of collecting information on registers and in memory fields, and so on.

The main purpose of the second macroroutine is to prevent unauthorized access of one user to information of another both during processing of their data in the combined mode, and during storage of this information in databases.

Protection of users from themselves (third macroroutine) assumes setting up conditions such that no possible errors of a user (neither accidental nor intentional) would lead to possibilities of information leakage on the part of the given user or any other.

The fourth macroroutine (protection of the system from itself) sets up secure barriers on the path of information leakage that arises as a consequence of errors in the main components of the automated control system (computing center) or as a consequence of malfunctions (accidental or malicious).

Specialized nature as a principle of security organization assumes two aspects: first--in view of the specific peculiarities of the problem under consideration, a reliable security mechanism can be planned and organized only by specialists trained in information security, and second--to ensure effective operation of the security mechanism, there must also be specialists in information security on the staff of the automated control system or computing center.

In virtue of the requirements of this principle, there has been considerable proliferation of specialized companies on different aspects of security outside of the USSR. For example, in the United States there are now about twenty comapnies specializing in security problems. The principal forms of services are consultative, conducting examinations and inspections, developing and installing means of protection.

Consultation is the simplest form of service, and involves acquainting clients with the essentials of the problem and possible approaches to solution; consultation takes the form of either conversations with individuals (most often the directors of the automated control system or computing center), or classes for a group of persons.

Examinations and inspections involve the company sending out a team of specialists to the client, who then study the conditions of operation of the automated control system or computing center, and evaluate the possibility and danger of information leaks. Based on this study, recommendations are made on the most effective organization of information security.

124

Development, sale and installation of security facilities is the most widespread form of service. Specialized companies develop and sell security program packages, physical and hardware means of security. Among physical means, the most widely used are intruder alarm systems, and among hardware means—data enciphering equipment.

The practice of setting up regular information security subdivisions (services) in automated control systems and computing centers is also widely used outside the Soviet Union. In the United States, the director of such a subdivision is called a security officer.

The principle of *informality* means that the methodology of designing the security mechanism and ensuring its operation is at base informal. This informality is interpreted in the sense that there is no engineering method of designing a security mechanism in the conventional sense of the term. The design methods that have been developed up until now contain sets of requirements, rules, a sequence and content of stages. Especially important is the fact that all these requirements, rules and content of stages are formulated on an informal level, i. e. mechanical implementation in the general case is impossible. However, it has been stated that satisfaction of requirements by information security specialists provides an objective basis for adequate security of the protection system.

At the present time, both general requirements for the security mechanism and requirements for its individual components have been fairly clearly formulated.

The general requirements for the security mechanism are as follows [Ref. 10]:

—adequacy, i. e. ensuring the required level of security (determined by the degree of secrecy of the information to be processed) with minimum expenditures on setting up the security mechanism and keeping it in operation;

—convenience for users, based on the requirement that the security mechanism should not create additional difficulties for users that require appreciable efforts to overcome them;

—minimization of access priveleges offered to users, i. e. each user should be given only really necessary rights of access to system resources and data;

—completeness of monitoring, i. e. obligatory monitoring of all referrals to protected data;

—punishability of violations; the most widespread means of punishment is refusal of access to the system;

—economy of the mechanism, i. e. minimum expenditures on setting up and operating the mechanism;

—design independence of secrecy, i. e. the security mechanism should function effectively even if its structure and content are known to an intruder.

125

A system including up to 200 special requirements has been developed in the plan of concretizing these general requirements in application to various components of automated control systems and computing centers with various operating conditions. For example, automated databanks should provide facilities for identifying users and resources of the system with periodic change of the identifying information, multiaspect delimitation of access to database components (by elements, by allowed procedures, by conditions of performing operations and so on), data encryption, registration of referrals to protected data, monitoring use of protected data, response to unauthorized actions (informing user, removing routine, disconnecting terminal, temporary delay in filling request, eliminating violator from user lists, transmission of alarm signal and the like).

Besides, convenience should be provided in specification of conditions and requirements of security, and the cost should be relatively low [Ref. 11].

The main requirement for protecting information that is to be transmitted over long distances on communication lines is that the secrecy of the transmitted data must be preserved even if they should reach an eavesdropper. The only means of reliable protection of information under these conditions is data encryption. The main requirements for encryption are:

--the difficulty of the enciphering and deciphering procedures should correspond to the degree of secrecy of the transmitted data;

--the resistance of the encipherment should be such that secrecy is not broken even in the case where the eavesdropper learns the cipher system;

--the cipher system and keys should not be overly complex;

--execution of ciphering and deciphering procedures should be as simple as possible;

--ciphering and deciphering procedures should not depend on the length of the message;

--ciphering and deciphering errors should not propagate through the system;

--message redundancy introduced by encipherment should be kept to a minimum.

Considerable attention has been given to selection, training, and work organization of personnel in automated control systems and computing centers. First of all, the initial selection of candidates for work in an automated control system or computing center is done with consideration of security requirements: a loyalty check is carried out, a recommendation is required from known organizations or persons and so on. Those signed up for work are acquainted with the security requirements, and those accepted for work are taught the rules of security. During work, personnel are kept under regular observation, and conformance with security rules is checked. In addition, conditions are brought about that stimulate adherence to these rules.

126

The general procedure of designing a security mechanism is usually divided into the following stages [Ref. 10]:

--analysis of structure and technology of operation of the automated control system or computing center for which the security mechanism is being developed;

--analysis of requirements for protection of the information to be processed in the automated control system or computing center;

--evaluation of information vulnerability;

--investigation and development of necessary security measures;

--estimate of necessary expenditures on organizing and providing the required security;

--development of rules and procedures for protecting information during operation of the automated control system or computing center;

--distribution of rights and duties on security, and definition of individual responsibility for security for each person in the staff of users and personnel of the automated control system (computing center);

--implementing the security mechanism plan;

--setting up conditions for conscientious attitudes toward observation of security measures;

--organizing reviews of the security mechanism.

The following are determined for each of these stages: list of specific questions to be answered; the one responsible for answering each of the questions; what practical steps must be taken; where should particular stress be placed; how to determine that work on the stage has been properly done.

Ref. 2 gives the possible content of a solution on the enumerated aspects for each design stage.

The consensus is that when these problems are handled by the given procedure with consideration of the above requirements, reliable information security can be provided in automated control systems and computing centers.

REFERENCES

1. Glushkov, V. M., "Problems of OGAS on the Current Stage", ALGORITMY I ORGANIZATSIYA RESHENIYA EKONOMICHESKIKH ZADACH, No 6, 1975, pp 5-14.

2. Gerasimov, V., Vladislavskiy, V., "Large-Scale Automation and Information Security", ZARUBEZHNAYA RADIOELEKTRONIKA, No 2, 1975, pp 49-63.

3. Hoffman, L. J., "Modern Methods of Information Protection", Moscow, Sovetskoye radio, 1980, 264 pages.

4. Gerasimov, V., Vladislavskiy, V., "Cryptographic Methods of Information Protection in Automated Systems", ZARUBEZHNAYA RADIOELEKTRONIKA, No 10, 1975, pp 53-68.

5. Kurmit, A. A., "Cryptographic Methods of Information Protection in Computer Systems", ZARUBEZHNAYA RADIOELEKTRONIKA, No 7, 1979, pp 17-41.

6. Gerasimov, V., Vladislavskiy, V., "Hardware Methods of Information Protection in Automated Systems", ZARUBEZHNAYA RADIOELEKTRONIKA, No 6, 1975, pp 34-46.

7. Gerasimov, V., Vladislavskiy, V., "Software Methods of Information Protection in Computers", ZARUBEZHNAYA RADIOELEKTRONIKA, No 8, 1975, pp 3-14.

8. Gerasimov, V., Vladislavskiy, V., "Organizational Measures of Information Security in Automated Systems", ZARUBEZHNAYA RADIOELEKTRONIKA, No 2, 1976, pp 53-67.

9. Gerasimov, V., Vladislavskiy, V., "Use of Security Methods and Means in Non-Soviet Automated Systems", ZARUBEZHNAYA RADIOELEKTRONIKA, No 4, 1976, pp 3-28.

10. Gerasimenko, V. A., Razmakhin, M. K., "Principles and Methods of Designing Information Security Mechanism in SEOD", ZARUBEZHNAYA RADIOELEKTRONIKA, No 5, 1981, pp 81-95.

11. Kazarov, M. S., "Information Security in Databases", ZARUBEZHNAYA RADIO-ELEKTRONIKA, No 12, 1979, pp 46-67.

12. Hemphill, Ch. F. (Jr.), Hemphill, J. M., "Security Procedures for Computer Systems", Homewood, Dow Jones-Irwin, Inc., 1973, 264 pages.

13. Hsiao, D. K., Kerr, D. S., Madnicc, S. E., "Computer Security", New York, Academic Press, Inc., 1979, 284 pages.

14. Martin, J., "Security, Accuracy and Privacy in Computer Systems", New York, Prentice-Hall, Inc., 1973, 640 pages.

6610
CSO: 1863/28   - END -